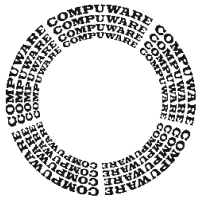
STROBE MVS

Concepts and Facilities

Release 3.0



COMPUWARE®

Please direct questions about STROBE MVS
or comments on this document to:

STROBE MVS Technical Support
Compuware Corporation
124 Mount Auburn Street
Cambridge MA 02138-5758
1-800-585-2802 or
1-617-661-3020
1-617-498-4010 (fax)
strobe-sup@compuware.com
Outside the USA and Canada, please contact
your local Compuware office or agent.

This document and the product referenced in it are subject to the following legends:

Copyright 2002 Compuware Corporation. All rights reserved. Unpublished rights reserved under the Copyright Laws of the United States.

U.S. GOVERNMENT RIGHTS-Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in Compuware Corporation license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable. Compuware Corporation.

This product contains confidential information and trade secrets of Compuware Corporation. Use, disclosure, or reproduction is prohibited without the prior express written permission of Compuware Corporation. Access is limited to authorized users. Use of this product is subject to the terms and conditions of the user's License Agreement with Compuware Corporation.

STROBE, iSTROBE, and APMPOWER are trademarks or registered trademarks of Compuware Corporation.

Acrobat® Reader copyright © 1987-1998 Adobe Systems Incorporated. All rights reserved. Adobe, Acrobat, and Acrobat Reader are trademarks of Adobe Systems Incorporated.

AD/Cycle, BookManager, CICS, DB2, IBM, IMS/ESA, Language Environment, MQSeries, OS/2, Software Mall, VisualGen, and VTAM are trademarks of International Business Machines Corporation. Microsoft is a registered trademark of Microsoft Corporation. Windows, Windows NT, and Windows 98 are trademarks of Microsoft Corporation. Attachmate and EXTRA! are registered trademarks of Attachmate Corporation. RUMBA is a registered trademark of Wall Data Inc. WRQ and Reflection are registered trademarks of WRQ, Inc.

Adobe ® Acrobat ® Reader copyright © 1987-2001 Adobe Systems Incorporated. All rights reserved. Adobe and Acrobat are trademarks of Adobe Systems Incorporated.

All other company and product names are trademarks or registered trademarks of their respective owners.

Doc. CWSTGX3A
January 27, 2003.

Contents

Figures

Summary of Changes

STROBE and this manual have changed from STROBE MVS for Sysplex Release 2.5.0. This section discusses the functional changes to STROBE, as well as the changes to this manual, from STROBE MVS for Sysplex Release 2.5.0 to Release 3.0.

Changes to STROBE

STROBE MVS for Sysplex Release 3.0 introduces:

- AutoSTROBE enhancements
  - AutoSTROBE SMF Candidate Utility-This tool enables you to scan System Management Facility (SMF) data sets to determine which job steps may require AutoSTROBE monitoring and measuring.
  - AutoSTROBE Autoloader- AutoSTROBE now will look at each batch job and its resource consumption as it ends. If the resource consumption is significant enough, the Autoloader will add the job step to the AutoSTROBE monitor candidate list.
  - STROBE - CHANGE AUTOSTROBE REQUEST panel is a new panel that allows you make changes to an AutoSTROBE request.
  - STROBE - AutoSTROBE STATUS panel provides an option to generate Performance Profiles.
  - Installation parameter to set minimum program execution time required to trigger AutoSTROBE basis data collection.
  - STROBE - ADD AUTOSTROBE REQUEST panel contains:
    - fields to disable AutoSTROBE action regardless of whether basis data thresholds are exceeded.
    - a field allowing you specify an ADD ACTIVE request, an ADD QUEUED request or both.
    - a field that allows you to set by individual AutoSTROBE request whether a measurement or warning is generated when thresholds are exceeded.
  - Installation parameter to set input values for AutoSTROBE algorithms that calculate thresholds
  - STROBE - STATUS panel identifies AutoSTROBE-initiated requests
- z/OS Memory Object Storage report detailing information about z/OS memory objects
- Java Virtual Machine (JVM) Support
  - New Java code measurement targeting capabilities through ISPF panels.
  - New Java reports showing new types of JVM Java (Just-In-Time JITted and interpreted) method measurement data at a summary and more detailed levels in batch, CICC and DB2 environments
- DDIO indexing-STROBE can use Compuware DDIO files in addition to map data sets as input to Performance Profile Report indexing.
- zOS V1R3 support

- IMS 8.1 support
- CICS Transaction Server Version 2.2 support
- ADABAS through and including V7.1.3 support
- Natural through and including Version 3.1.5 support
- MQSeries Release 5.3 support
- IBM C/C++ Releases for OS/390 V2R9, V2R10 and IBM C/C++ for z/OS V1.1 through V1.4 support

Changes to this Manual

This section discusses the changes to this document for STROBE MVS for Sysplex Release 3.0.

Chapter 3

- An example and description of the z/OS Memory Objects report has been added.
- The descriptions of report fields that may show DDIO indexing information have been updated

Introduction

This manual is a guide to application performance management (APM) using the STROBE MVS Application Performance Measurement System, a product designed for use on IBM MVS/ESA, IBM OS/390, and IBM z/OS systems. It describes the benefits of using STROBE and explains its capabilities. It describes the set of reports that make up the STROBE Performance Profile reports, and presents a guide to interpreting the information shown in the Performance Profile.

How This Manual Is Organized

Chapter 1, "Overview" describes the benefits of APM and presents an overview of the STROBE system.

Chapter 2, "Interpreting the STROBE Performance Profile" presents a general discussion of how to interpret the reports in the Performance Profile and then presents a case study with reports generated from a sample COBOL program.

Chapter 3, "The STROBE Performance Profile" describes each report in the STROBE Performance Profile. It explains report terminology and discusses fields within each report.

Appendix A, "Program Structure and STROBE Pseudo-Entities" presents a sample program structure and discusses the system modules whose activity STROBE compresses to make reports more concise.

Appendix B, "Example Program and STROBE Performance Profile" presents a sample COBOL program and the STROBE Performance Profile that resulted from measuring and indexing the program.

How to Use This Manual

To understand the benefits of STROBE and to determine which programs to measure, read Chapter 1, "Overview" before you submit a measurement request. For help in interpreting a STROBE Performance Profile, read Chapter 2, "Interpreting the STROBE Performance Profile" and Chapter 3, "The STROBE Performance Profile".

The STROBE Library

The STROBE base product manuals include:

- *STROBE MVS Concepts and Facilities*, document number CWSTGX3A

  STROBE MVS Concepts and Facilities explains how to decide which programs and online regions to measure, when to measure them, and how to interpret the reports in the STROBE Performance Profile.

- *STROBE MVS Messages,* document number CWSTXM3A

  STROBE MVS Messages lists all messages and abnormal termination (ABEND) codes, describes how to interpret them, and in many cases suggests a corrective action.

- *STROBE MVS System Programmer's Guide*, document number CWSTXI3A

The *STROBE MVS System Programmer's Guide* explains how to install and maintain STROBE.

- *STROBE MVS User's Guide*, document number CWSTUX3A and the *STROBE MVS User's Guide with Advanced Session Management*, document number CWSTUA3A

  The *STROBE MVS User's Guide* explains how to use STROBE to measure application performance. The *STROBE MVS User's Guide with Advanced Session Management* explains how to use STROBE with the STROBE Advanced Session Management Feature to measure application performance. Users who have the STROBE Advanced Session Management Feature will use this manual rather than the *STROBE MVS User's Guide*.

- *STROBE MVS Application Performance Measurement System Quick Reference*

  The *STROBE MVS Application Performance Measurement System Quick Reference* is a convenient reference for how to use STROBE and for interpreting the STROBE Performance Profile.

STROBE Feature Manuals

These manuals describe the optional features of the STROBE MVS Application Performance Measurement System. Each manual describes measurement concepts applicable to and specific data made available by the feature.

- *STROBE MVS User's Guide with Advanced Session Management*, document number CWSTUA3A
- *STROBE ADABAS/NATURAL Feature*, document number CWSTUN3A
- *STROBE CA-IDMS Feature*, document number CWSTUR3A
- *STROBE CICS Feature*, document number CWSTUC3A
- *STROBE COOL:Gen Feature*, document number CWSTUG3A
- *STROBE CSP Feature*, document number CWSTUP3A
- *STROBE DB2 Feature,* document number CWSTUD3A
- *STROBE IMS Feature,* document number CWSTUI3A
- *STROBE Interface Feature*, document number CWSTUF3A
- *STROBE Java Feature,* document number CWSTUJ3A
- *STROBE MQSeries Feature,* document number CWSTUM3A
- *STROBE UNIX System Services Feature,* document number CWSTUU3A

Online Documentation

STROBE manuals are available in HTML, Adobe Acrobat PDF format, and IBM BookManager format, on CD-ROM and at Compuware's technical support Web site at http://frontline.compuware.com.

Online Help

STROBE products provide the following online information:

- STROBE/ISPF Online Tutorials, Option T from the STROBE/ISPF STROBE OPTIONS menu
- STROBE/ISPF Online Message Facility, Option M from the STROBE/ISPF STROBE OPTIONS menu.

Other Compuware Application Performance Management Products

The following products and features work in conjunction with the STROBE MVS Application Performance Measurement System. These tools extend the benefits of application performance management (APM).

iSTROBE

iSTROBE enables you to view and analyze STROBE Performance Profile data on a workstation using a standard Web browser. Easy to install and easy to use, iSTROBE guides you through the performance analysis process and offers recommendations for improving performance. iSTROBE simplifies the performance analysis of applications that you measure with STROBE. For more information on iSTROBE, see the *iSTROBE Getting Started Guide*.

SQL Analysis Feature

The SQL Analysis Feature works in conjunction with STROBE and iSTROBE or APMpower to supply access path analyses and database and SQL coding recommendations for DB2 applications measured by STROBE. The SQL Analysis Feature pinpoints the most resource-consumptive static or dynamic SQL statements, explains why these statements might be inefficient, and provides recommendations to improve the performance of the DB2 application. For more information on the SQL Analysis Feature, see the *STROBE MVS User's Guide* or the *STROBE MVS User's Guide with Advanced Session Management*.

APMpower

The APMpower Application Performance Analysis System extends the benefits of STROBE to application developers who use workstations to develop, test, and maintain MVS applications. Developers employ the APMpower graphical user interface and advanced analytical aids to navigate the Performance Profile, analyze and improve application performance, and share performance knowledge across the IS organization. For more information about APMpower, see the APMpower documentation.

Compuware APM Technical Support

For North American customers, for technical support, please contact the Technical Support department by telephone at (800) 585-2802 or (617) 661-3020, by fax at (617) 498-4010, or by e-mail at strobe-sup@compuware.com.

To access online technical support, visit Compuware's FrontLine page on the World Wide Web at http://frontline.compuware.com and select the product "STROBE and APMpower."

For other international customers, please contact your local Compuware office or STROBE supplier.

Compuware APM Training

Compuware's Education Resources Group offers a range of training options for organizations that use STROBE, iSTROBE, and APMpower. To arrange Application Performance Management training, please contact Compuware at 1-800-835-3190 or visit Compuware's Education Resources Group at http://www.compuware.com/training

For other international customers, please contact your local Compuware office or STROBE supplier for a complete list of APM Training offerings.

Compuware APM Service Offerings

For North American customers, for information about current service offerings, please contact your local Compuware sales office or call Compuware Corporate Headquarters at 1-800-COMPUWARE (266-7892) or visit Compuware's APM Product page on the World Wide Web at http://www.compuware.com/products/strobe.

For other international customers, please contact your local Compuware office or STROBE supplier for a complete list of Services offerings.

APM Installation Assurance

The APM Installation Assurance service assists you in planning for, installing, customizing and using APM products. The service will help you maximize the value and benefits derived from the APM product family.

Consulting engineers work closely with your IT personnel to understand your operating environment and your organization's APM goals. The engineer will assist you in developing a customization and installation plan for STROBE, iSTROBE, and APMpower. The engineer will oversee the installation process and verify product readiness. The engineer will also help set up measurement request schedules, request groups, history records, AutoSTROBE measurement requests, and will verify the installation of the SQL Analysis Feature.

With APM Installation Assurance services, your organization can immediately maximize the value received from your investment in the APM product family. You will also benefit from a fully customized installation that will enhance the product functionality and increase the automation aspects of your APM initiatives.

Application Performance Management Consulting

The Application Performance Management (APM) Consulting services assist you in identifying and resolving specific performance problems in your OS/390 business-critical applications.

Using STROBE, iSTROBE, and APMpower, consulting engineers work closely with your IT personnel to measure an application's performance, identify performance improvement opportunities and make recommendations for implementing solutions.

With APM Consulting services, your organization cannot only resolve problems quickly and effectively, but also gain the skills necessary to prevent application performance degradation in the future.

Application Performance Assessment

The Application Performance Assessment (APA) service assists you in achieving a higher level of performance for your OS/390 business-critical applications.

Using STROBE, iSTROBE, and APMpower, consulting engineers work closely with your IT personnel to evaluate the efficiency of business-critical applications, identify opportunities for improving performance and document the potential savings that can result from implementing recommended solutions.

With APA services, you cannot only improve application performance quickly and effectively, but also gain the knowledge and skills necessary to implement and sustain a process-oriented application performance management (APM) program.

Chapter 1. Overview

STROBE is a software product that reports on how your application programs use resources in IBM MVS/ESA and IBM OS/390 environments. It determines where and how an online or batch application spends time. Incorporating STROBE measurement into each phase of the application life cycle—design and development; build, test and quality assurance; production; and maintenance—ensures that your applications are designed to run efficiently and responsively and that no performance problems are unintentionally introduced.

STROBE employs a sampling technique that executes within the address space of the target application and periodically takes snapshots of an application's execution. STROBE stores this data in a *sample data set*, and organizes the information to produce the STROBE *Performance Profile*, a hierarchical series of reports that show where and how time was spent during the application's execution. The Performance Profile can indicate which few lines of code, among perhaps thousands, are using significant amounts of system resources. With this focus, you can examine your coding techniques and make changes to only those areas of code that have a major effect on performance.

The Benefits of APM with STROBE

STROBE and APMpower enable you to implement an *application performance management* (APM) program, the quality management discipline used by IS professionals to deliver efficient and responsive applications and to maintain high levels of application performance throughout an application's life cycle. APM ensures that each functional group within the IS organization views application efficiency and responsiveness as essential quality attributes of the products it delivers.

Efficient and responsive application performance is paramount to meeting the business goals of an enterprise. The cost of inefficiencies in business-critical applications can be measured in lost revenue, wasted resources, and weakened competitive position. An APM strategy establishes IS standards through which you can evaluate and control an application's performance as the application moves through its life cycle. The benefits of APM include developing and maintaining efficient applications, delivering responsive applications, increasing system capacity for additional applications or program development, and controlling costs.

For each phase of the application life cycle, APM defines a set of methods, benchmarks, and tasks that ensure efficient and responsive applications. In an APM discipline, products specific to application performance—STROBE MVS and APMpower—are used with other software products to optimize application performance.

With the STROBE Advanced Session Management Feature you can also collect and display historical information for measurement sessions. With this measurement session history, you can more easily identify and track changes in the performance of a particular job step, online region transaction, or DBRM and relate these changes in performance to changes in operating costs. This capability can be useful in all phases of application performance management. For more information, see the *STROBE MVS User's Guide with Advanced Session Management*.

The following sections describe how to incorporate STROBE in each phase of the application life cycle.

Design and Development

As a development professional, you can use STROBE frequently during application development. Measuring with STROBE in development enables you to identify and fix performance problems early in the application's life cycle. When designing a new application, you can measure the new code with STROBE to ensure that it is efficient in its use of system resources. As you develop an application, you can periodically measure with STROBE to identify and eliminate any performance problems inherent in the design before your code enters the test stage.

You can assess alternative programming techniques by measuring the performance of various test cases. This approach is useful for evaluating the resource demands of generated code (4GL), the relative efficiency of algorithms, and the effect on resource usage of such factors as data type. With STROBE you can also determine the relative resource demands of access methods in response to different parameters.

Build, Test, and Quality Assurance

You can incorporate STROBE into the build, test, and quality assurance process. With STROBE you can confirm design assumptions about a new application's resource requirements by measuring the application when it is exposed to production volumes of data. You can measure the resource use of a new release of an application and compare it with the production release it is to replace; this comparison will ensure that inefficiencies have not been unintentionally introduced and that program changes do not degrade performance.

You can embed STROBE measurement into test jobs, making performance measurement part of the test procedure. Your test cycle then ensures that your application is responsive and efficient.

Production

You can use STROBE in a systematic effort to improve the general efficiency of your application portfolio and reduce costs. By reducing the resource use of those applications that place the highest demands on your system resources, you free resources for other applications. Once you have investigated and acted on all the performance improvement opportunities in an application, you can develop a set of performance benchmarks for later measurement and improvement. You can plan the resource load of the application and respond early if a new release fails to meet your benchmarks.

Maintenance

During program maintenance, programmers make changes to existing code—whether to fix a problem, enhance application function, or refit the application to run in a new operating environment. Using STROBE in maintenance helps ensure that application efficiency continues to conform to established benchmarks.

When you first implement an APM discipline, you are likely to discover substantial opportunities for improving the general efficiency of your data center, enhancing service, and reducing processing costs. During maintenance, you can employ STROBE in a systematic effort to reclaim system resources used by inefficient applications. For example, you can free resources for additional processing by targeting those batch processing programs and online applications that place the highest demand on the system.

An Overview of STROBE

Using STROBE within an APM discipline is an iterative process. Your first step is to decide what program to measure. You then submit a request for measurement and produce a

Performance Profile from the information gathered. You interpret the Performance Profile, focusing on high concentrations of CPU activity or elapsed time in your programs, or in service routines invoked by your application. As a result of your interpretation, you make improvements to your source code and then measure again with STROBE to verify your results and, possibly, to reveal additional performance improvement opportunities. This process is diagramed in Figure 1-1 on page 1-4.

The flexibility and powerful reporting capabilities of STROBE make it a convenient and useful product. STROBE provides you with a number of options, such as:

- selecting job steps for measurement
- controlling the duration and rate of measurement
- controlling the type and amount of data collected
- determining where STROBE saves the data

For more detailed instruction on the steps discussed next, see the *STROBE MVS User's Guide* or the *STROBE MVS User's Guide with Advanced Session Management*.

What to Measure

When considering which programs to measure, focus on those that are the heaviest users of CPU or experience elapsed time that is significantly above average. Particularly good candidates for improvement are programs whose CPU usage or elapsed time has recently increased.

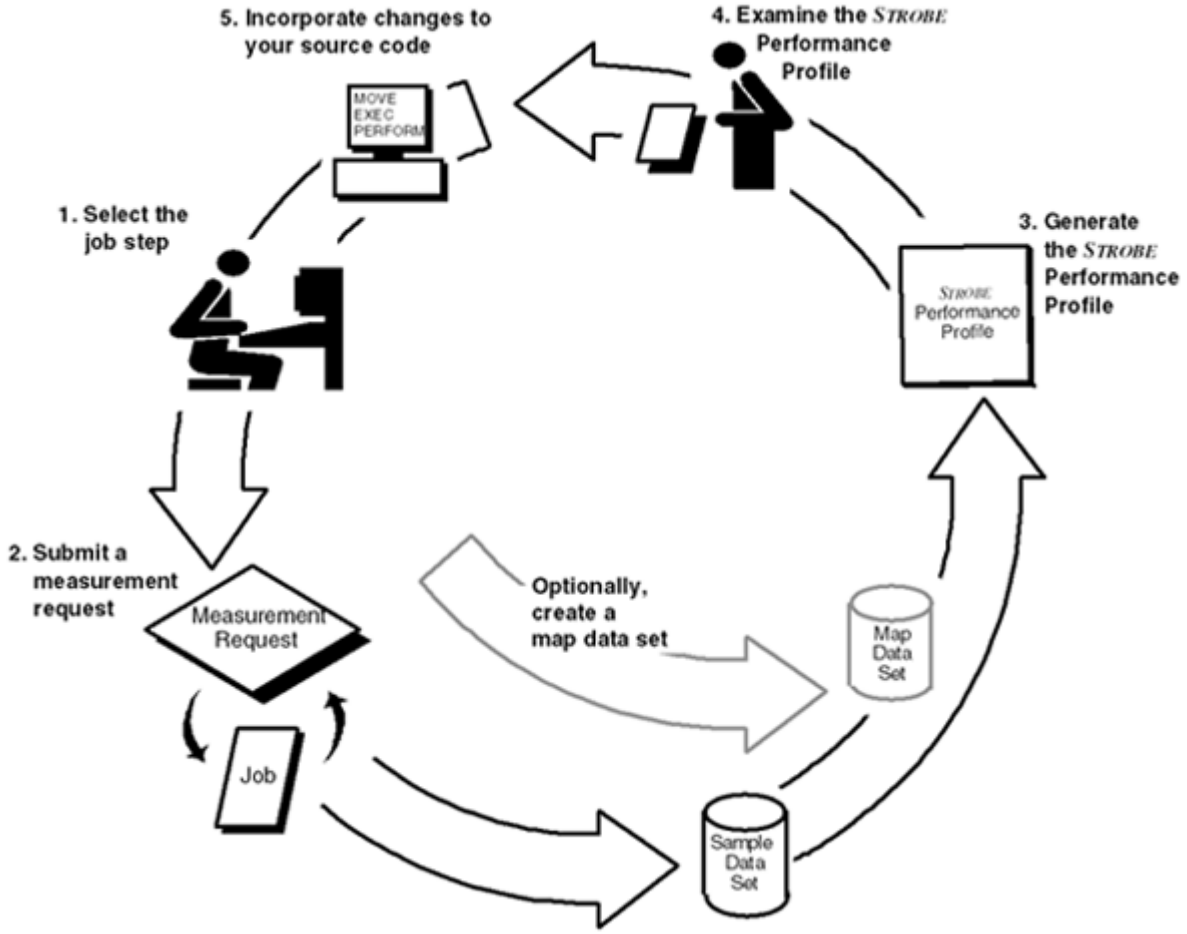
The level of CPU or elapsed time that you consider high, and therefore a candidate for improvement, depends on your objectives and standards. If you want to delay a hardware upgrade, any opportunity to save CPU time is important. If your data center charges back for CPU time, any reduction in CPU time represents a cost savings for your department. If, however, your online applications risk coming up late because of long-running application programs, excessive elapsed time becomes your focus. Often, the most significant cause of an application's extended elapsed time is wait time—the portion of run time during which the application was in the wait state when no task within the measured address space was able to make use of the CPU time available to it.

AutoSTROBE Measurement Candiates

Rather than determining yourself what applications you think are STROBE measurement candidates, you can use the Advanced Session Management AutoSTROBE Feature to help you select them. AutoSTROBE can screen and montior all of your system program activity and then based on upon certain performance criteria, initiate a measurement request for any job showing performance abnormality.

SeeChapter 6 of the *STROBE MVS User's Guide* with Advance Session Management for information about selecting STROBE measurement candidates with AutoSTROBE.

Figure 1-1. An Overview of Using STROBE



Invoking STROBE

Once you have selected a job step to measure, you can invoke STROBE through

- STROBE/ISPF—a convenient menu-driven interface to the STROBE system
- STROBE command language— a program that you can invoke through TSO commands or batch jobs
- APMpower—a workstation product that enables you to measure MVS applications, analyze performance data, and share application performance knowledge

The job step of the application that you decide to measure is a *target job step*. STROBE offers a way to

- submit a request to measure the job step, whether currently executing or scheduled to run later
- retrieve the results of the measurement and produce reports

Managing Measurement

STROBE identifies the period of time it collects data for a sample data set as a *measurement session*. A *measurement request* initiates one or more measurement sessions, with an equal number of sample data sets.

When you submit a request to measure a target job step, STROBE manages the request by

- storing and tracking it
- initiating a measurement session to service it
- associating the details of the measurement session (including the name of the sample data set and messages resulting from it) with the measurement request

When you submit a measurement request, you must identify the name of a job and the job step that you want to measure. You can also specify measurement session parameters, such as the target sample size, measurement session duration, and sample data set information. You can instruct STROBE to collect the types of data you need for your analysis, although in most instances STROBE collects the right kind of data about the application by default. Optionally, you can request that STROBE automatically creates a Performance Profile upon completion of the measurement session.

For more information on measurement parameters, see Chapter 2, "Interpreting the STROBE Performance Profile" of the *STROBE MVS User's Guide* or *STROBE MVS User's Guide with Advanced Session Management*.

During execution of the target job step, STROBE uses a sampling technique to collect data about the system resources used within an active job step. At set intervals, STROBE interrupts the executing job step to record various types of information about the job that you are measuring.

STROBE provides flexibility for measuring different types of applications. You can manage STROBE measurement requests that target batch processing applications, online applications, and multiregion online applications.

Batch Processing Applications

You can use STROBE to investigate a critical batch processing program in production that takes much longer than expected to complete. You can invoke STROBE to initiate measurement while the job is being processed. If the production batch processing program executes when you are not logged on, you can submit a STROBE measurement request before the job is processed. When the targeted production job step is dispatched, STROBE begins measuring.

Online Applications

You can measure online applications in response to changing transaction loads. When you measure an online application, you can conduct a number of measurements during the application's most active periods rather than one long measurement.

In addition, STROBE enables you to respond to online activity while you observe it. You can control a measurement session while it is in progress by issuing commands to STROBE. You can:

- start or restart sampling
- suspend or stop sampling
- switch to a new sample data set
- terminate measurement

If you schedule measurement during peak periods, you can detect trends in resource growth that provide information to help you in capacity planning. You can limit sampling to those periods in which you are interested.

Short measurement sessions can help you, if you are responsible for data processing services, to diagnose operational problems within an online region. A brief measurement session can help you identify the cause of performance problems such as sluggish response, saturation of the CPU, saturation of a channel, or excessive storage demand.

Long measurement sessions can help you to obtain information about resource use over time. You can conduct extended sessions to determine the resource demand of individual transaction types or application programs on the system.

Multiregion Online Applications

Some applications use more than one online subsystem (such as CICS, DB2, CA-IDMS or IMS) consisting of one or more service regions and one or more dependent regions, each executing in its own independent address space. You can conduct independent measurements in all of these address spaces. With the STROBE Advanced Session Management Feature, you can create a group of measurement requests that will simultaneously measure activity in multiple address spaces. For example, you can measure an IMS application that processes data managed by DB2. You can then see a more complete picture of the application's overall performance by reviewing the Performance Profiles for both the IMS and the DB2 regions.

Producing the Performance Profile

When measurement is completed and the sample data set is closed, you submit a command through STROBE/ISPF or STROBE command language to produce a Performance Profile, or you can request that STROBE automatically produce the Performance Profile for you, as part of the measurement request. Alternatively, you can produce a Performance Profile by executing a batch job using Compuware-supplied procedures. The Performance Profile is a series of reports that associates the performance information in the sample data set to sources within your application code. STROBE relates information in the sample data set to the hexadecimal offsets in your application, showing you where the application uses resources. If you perform the step called *indexing*, STROBE associates the sample data set's measurement information with your application's line numbers and procedure names.

Indexing

When measuring, STROBE can identify the offset of the instruction from the beginning of the executing module, but cannot identify source code statements or statement numbers unless they are present during execution. *Indexing* is an optional STROBE activity that uses the listing from a source language compiler to identify procedures within a source module. When you index a program, STROBE creates a *map data set* that relates the source code statements and line numbers in the program to the offsets in the compiled object module. You then produce a Performance Profile using the map data set as well as the sample data set as input. This Performance Profile is referred to as indexed; that is, it relates activity to your application's line numbers and procedure names instead of the hexadecimal offsets.

Indexing and the language-specific rules that govern indexing are detailed in "Indexed Performance Profiles" on page 2-26.

STROBE Indexing Support

The STROBE system includes model cataloged procedures that compile and index source programs written in the following languages and development environments:

- AD/Cycle C/370
- AD/Cycle COBOL/370
- AD/Cycle PL/I
- ADS/O
- assembler language
- CA-IDMS dialogs
- CA-Optimizer
- COBOL
- COBOL 370
- COOL:Gen (formerly Composer)

- CSP
- DB2 SQL
- FORTRAN
- IBM C/370
- IBM C/C++ for MVS/ESA
- IBM OS/390 C/C++
- IBM COBOL for OS/390 and VM
- IBM COBOL for MVS and VM
- VisualAge COBOL for OS/390
- IBM PL/I for MVS and VM
- NATURAL
- PL/I
- SAS/C
- VisualAge PL/I

Note: You can also use DDIO files for indexing. DDIO is a Compuware Shared Services (CSS) file access method. If you have created DDIO files that contain either COBOL or PL/I listings, these can be selected to cause the Performance Profiles generated for the COBOL or PL/I application to include DDIO file indexing. See"Indexing Source Modules from DDIO Files" on page 3-21 of the *STROBE MVS User's Guide with Advanced Session Management* for more information about indexing and DDIO files.

Interpreting the Performance Profile

Your interpretation of the reports in the Performance Profile points you to opportunities to reduce CPU time or wait time. Chapter 2, "Interpreting the STROBE Performance Profile" provides an example of a Performance Profile and explains how to interpret the reports. Chapter 3, "The STROBE Performance Profile" provides a sample of each report in the Performance Profile, describing all the fields and columns.

STROBE Features

Additional subsystem and language-specific performance data is available in your Performance Profile if you are using STROBE Features. The STROBE Features extend your ability to manage application performance by providing additional facilities to collect specialized data about system performance and assisting your analysis of applications compiled in a variety of program languages.

Chapter 2.
Interpreting the STROBE Performance Profile

This section outlines the general steps to follow when you are interpreting a STROBE Performance Profile. It then presents a case study that exemplifies how to analyze a Performance Profile to investigate and improve the performance of an application program that has recently experienced a dramatic increase in wait time.

The final section guides you in interpreting indexed reports by explaining in detail the process of indexing and by listing the language-specific rules that govern the appearance of indexed reports.

The Performance Profile provides a comprehensive picture of your application's resource use. Each report concentrates on one aspect of performance. For example, the Most Intensively Executed Procedures report identifies the most heavily used procedures, and the Wait Time by Module report shows each module and section name in which STROBE found the target program to be in the wait state.

STROBE Performance Profile Interpretation Steps

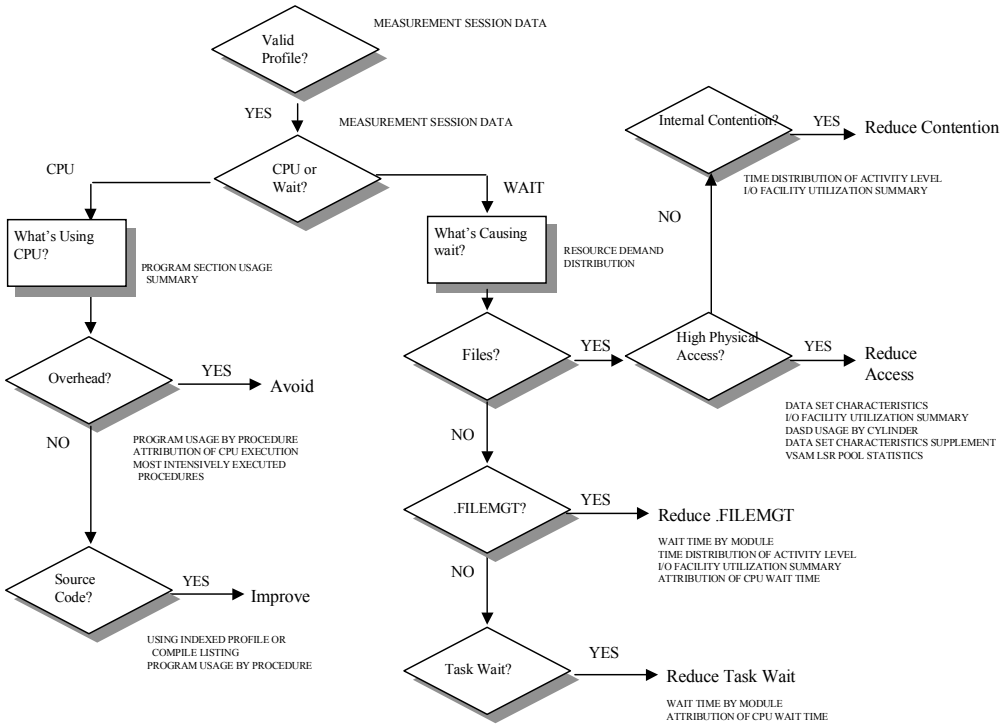
The Performance Profile begins with the Measurement Session Data report and is structured hierarchically so that subsequent reports provide detail for the higher-level reports. This section discusses the steps you take to interpret the Performance Profile. The process is illustrated in the Performance Profile interpretation flowchart, Figure 2-1 on page 2-2. This flowchart details the interpretation path for batch applications. For online applications, follow the same path but also refer to the subsystem-specific reports that are described in the appropriate feature guide.

The flowchart indicates:

- major decision points with diamonds
- questions with boxes
- related reports to the right of the decision point or question

The flowchart indicates which reports in the Performance Profile help you to make a decision or answer a question. For example, the first decision is to determine whether the Performance Profile is valid, and subsequently whether to pursue a reduction in CPU time expenditure or wait time. Use the information in the Measurement Session Data report to make both of these decisions. Continue following the flowchart until you have determined the cause of high wait time or high CPU time.

Figure 2-1. STROBE Performance Profile Interpretation Flowchart



Samples of each of the reports are in Chapter 3, "The STROBE Performance Profile", which provides a reference for the columns and fields that appear on all the reports. The reports in the Performance Profile print in the order found in Chapter 3, "The STROBE Performance Profile", not in the order in which this chapter discusses them.

Is This a Valid Profile?

The first tasks in interpreting a Performance Profile are to verify that the reports in the Profile present measurement of the correct program on the correct system and that they are statistically valid.

To determine validity, refer to the Measurement Session Data report (Figure 2-2 on page 2-3). First, verify these fields to ensure that the values reflect the application job step that you intended to measure:

In the JOB ENVIRONMENT section, check:

- PROGRAM MEASURED
- JOB NAME
- JOB NUMBER
- STEP NAME
- CONDITION CODE, which appears only if the step ended while STROBE was measuring
- SYSTEM

In the MEASUREMENT PARAMETERS section check the options that you specified:

- ESTIMATED SESSION TIME.
- TARGET SAMPLE SIZE.

- REQUEST NUMBER, preceded by either a (Q) or an (A). "(Q)" indicates that you submitted a queued request, while "(A)" indicates that you submitted an active request.

Figure 2-2. Measurement Session Data Report

```
                                                ** MEASUREMENT SESSION DATA **

------- JOB ENVIRONMENT --------        ------- MEASUREMENT PARAMETERS ------        ----- MEASUREMENT STATISTICS ----

PROGRAM MEASURED   -    PAYROLLO        ESTIMATED SESSION TIME  -      45 MIN        CPS TIME PERCENT -          10.51
JOB NAME           -    WPAKMOGO        TARGET SAMPLE SIZE      -      10,000        WAIT TIME PERCENT -         89.49
JOB NUMBER         -    JOB01541        REQUEST NUMBER   (Q)    -          36        RUN MARGIN OF ERROR PCT  -    .97
STEP NAME   -             RUNAPP                                                    CPU MARGIN OF ERROR PCT  -   2.99

DATE OF SESSION    -  05/12/1999        SYS REQ - ALL SYSTEMS                       TOTAL SAMPLES TAKEN -      10,248
TIME OF SESSION    -    11:25:53                                                    TOTAL SAMPLES PROCESSED -  10,248
CONDITION CODE     -      C-0000        MODULE MAPPING BASELINE -           2       INITIAL SAMPLING RATE-   3.70/SEC
                                        BASELINE OVERRIDE -           TAXRTNO       FINAL SAMPLING RATE  -   3.70/SEC
SYSTEM -             ESA SP5.2.0        LIBRARY -             WPAKMO.USER.LOAD
DFSMS              -       1.2.0                                                   SESSION TIME -   52 MIN  5.98 SEC
CPU MODEL          -   3090-600S                                                   CPU TIME -        5 MIN 35.50 SEC
SMF/SYSTEM ID -        MVST/TLP01        --------- REPORT PARAMETERS ---------      WAIT TIME    -   45 MIN 53.92 SEC
                                                                                   STRETCH TIME -    0 MIN 36.56 SEC
REGION SIZE BELOW 16M  -  4,160K        REPORT RESOLUTION    -       64 BYTES
REGION SIZE ABOVE  -     32,768K        SORTSIZE             -      999,999        SRB TIME     -    0 MIN 17.40 SEC
                                        LINES/PAGE           -           60        SERVICE UNITS-            334494
PTF LVL- 2.3.0.FS000000/FS000000
                                        DASD= 2.0%  DASDGAP= 5                     PAGES IN-      0     OUT-       0
                                                                                   PAGING RATE  -           0.00/SEC
                                        DATE FORMAT               MM/DD/YYYY       EXCPS -     38,093      12.19/SEC
                                        TIME FORMAT (24 HOURS)      HH:MM:SS

SAMPLE DATA SET  - WPAKMO.PAY01.NEWGO.WPAKMOGO.S001D001.DATA
```

Check the margin of error fields in the MEASUREMENT STATISTICS section against the criteria outlined below. Because STROBE intermittently samples application execution rather than continuously monitoring, there is always some inaccuracy inherent in the measurement. Normally, this slight inaccuracy should not affect the reliability of the reports. STROBE quantifies this amount for reports that detail CPU time and for reports that detail run time.

- RUN MARGIN OF ERROR PCT, the margin of error for the percentages in the reports that detail run time. A value less than or equal to 2.00 indicates that the information in the Performance Profile is statistically valid. The percentages in reports that detail run time are reliable within a range of plus or minus this percentage.
- CPU MARGIN OF ERROR PCT, the margin of error for the percentages in the reports that detail CPU time. This value shows the margin of error for CPU execution samples. A high value in this field, over 10, indicates that the information in the reports that show program execution is most likely not valid and that you probably have not collected enough execution samples. If your goal is to reduce wait time, however, a high CPU margin of error does not affect the validity of the reports that show run time.

After you have determined the validity of the Performance Profile, pursue the steps for CPU time reduction, wait time reduction, or both. Many of the reports have histograms where a *spike*—the longest row of pluses and asterisks—clearly indicates observed high CPU time consumption or run time.

Wait or CPU?—Determining Performance Improvement Opportunities

Once you know that the Performance Profile is representative and valid for the job step that you intended to measure, determine whether reducing wait or reducing CPU time presents the best opportunity for improving performance. The Measurement Session Data report quantifies the time the application is using CPU resources and experiencing wait. Note the following on the Measurement Session Data report:

- SESSION TIME—the amount of time the session ran.

- CPS TIME PERCENT—the percentage of time that STROBE observed the application consuming CPU time. This value is calculated from the percentage of samples in which STROBE observed CPU execution.
- WAIT TIME PERCENT—the percentage of time STROBE observed the job step waiting for an event to complete.
- CPU TIME—the amount of time the application was using CPU resources.
- WAIT TIME—the amount of time STROBE observed that the central processing system was available but was not in use by application tasks executing within the measured job step.

Depending upon which quantity is larger and your goal in implementing an APM program, pursue an opportunity to reduce CPU time or wait time. For example, if your immediate concern is about a shrinking batch window, the first objective is to reduce wait time. If, however, costs related to CPU time are your main concern, investigate CPU time first. This section looks at the path to follow when looking for an opportunity to reduce wait time.

What's Causing Wait?

If wait time seems high, the Resource Demand Distribution report, Figure 2-3 on page 2-4, helps you determine whether files, tasks, or file management activity (.FILEMGT on the flowchart) are causing the wait. When STROBE detects file-processing activity that it does not relate to a specific ddname, it groups all the activity under the name .FILEMGT.

Figure 2-3. Resource Demand Distribution Report

** RESOURCE DEMAND DISTRIBUTION **

TASK OR DDNAME	RESOURCE	PERCENT OF RUN TIME SERVICED BY CPU	PERCENT OF RUN TIME SERVICED BY I/O	PERCENT OF RUN TIME SERVICED BY EITHER	PERCENT OF RUN TIME SPENT SOLO IN CPU	PERCENT OF RUN TIME SPENT SOLO IN I/O	PERCENT OF RUN TIME SPENT SOLO IN EITHER	PERCENT OF RUN TIME SPENT CAUSING CPU WAIT	CUMULATIVE PERCENTAGES SOLO TIME	CUMULATIVE PERCENTAGES CAUSING CPU WAIT
PAYROLL0	CPU	1.22	.00	1.22	1.21	.00	1.21	.24	1.21	.24
.FILEMGT		8.58	.89	9.46	8.57	.88	9.46	80.47	10.67	80.71
PAYCHECK	3380	.00	2.74	2.74	.00	2.74	2.74	2.98	13.41	83.69
TAXDATA	3380	.67	1.42	2.10	.67	1.41	2.09	2.40	15.50	86.09
EMPLOYEE	3380	.04	1.95	1.99	.04	1.95	1.99	3.33	17.49	89.42
SYS00010	3380	.00	.07	.07	.00	.07	.07	.07	17.56	89.49

The Resource Demand Distribution report identifies, in the CAUSING CPU WAIT column under PERCENT OF RUN TIME SPENT, the percent of run time that the tasks, .FILEMGT, or ddnames were waiting for an event (usually I/O) to complete.

Locate high values in this column and then follow the appropriate branch or branches of the flowchart. Often, the Resource Demand Distribution report shows some file wait, some task wait, and some wait related to .FILEMGT. You may need to follow more than one path on the flowchart to explore all the possible opportunities for improvement.

Are Files Causing Wait Time?

If the Resource Demand Distribution report shows a high value for files in the CAUSING CPU WAIT column, determine why the files are causing wait. Explore whether the Performance Profile relates this wait to high physical access or internal contention, or whether it suggests the possibility of external contention.

Is High Physical Access Causing Wait Time?

The Resource Demand Distribution report indicates that files are causing wait when high percentages appear for ddnames in the CAUSING CPU WAIT column under the PERCENT OF RUN TIME SPENT heading. If files are causing wait, read the Data Set Characteristics report (Figure 2-4 on page 2-5) to identify the physical characteristics of the files. Here

you may find that options such as the default values for block sizes and buffering are creating undue wait. For files with access methods such as VSAM, the Data Set Characteristics Supplement report (Figure 2-5 on page 2-6), and the VSAM LSR Pool Statistics report (Figure 2-6 on page 2-7), which detail buffering, provide important information.

VSAM and QSAM offer various options in blocking and buffering. While a comprehensive explanation of the tuning options is beyond the scope of this manual, the following paragraphs show how STROBE can indicate the blocking and buffering options you should consider modifying. For more detailed information relating to your specific Performance Profile, see the IBM *MVS/ESA SML Storage Management Reader's Guide* to find the appropriate manual.

QSAM Performance Improvement Opportunities

One possible opportunity to improve performance of QSAM data sets is to increase the block size of the file so that it is the maximum possible for the DASD you are using without wasting DASD space. Increasing the block size enables you to minimize the number of blocks transferred, represented on the Data Set Characteristic report as EXCP (direct invocations of execute channel programs) COUNTS, and significantly reduce wait time. Each time MVS issues an EXCP, the address space for the application program typically waits until the EXCP completes. Higher block sizes reduce the number of EXCPs and, consequently, the wait time.

Figure 2-4. Data Set Characteristics Report

```
                                        ** DATA SET CHARACTERISTICS **

DDNAME     ACCESS    POOL  REC  BLK/CI    HBUF  BUF  RPL  -SPLITS-     EXCP DATA SET NAME
           METHOD      NO  SIZE  SIZE       NO   NO STRNO CI    CA   COUNTS

EMPLOYEE VSAM KSDS          150  4096              2   1               7200 WPAV.WPAKMO.PAY01.EMPLOYEE
EMPLOYEE VSAM INDX         4089  4096              1                  13875 WPAV.WPAKMO.PAY01.EMPLOYEE
PAYCHECK QSAM               132  1320              5                  16800 WPAKMO.PAY01.OUTPUT.PAYCHEKS
STEPLIB                                                                   6
TAXDATA  VSAM KSDS           10  4096              2   1                 67 WPAV.WPAKMO.PAY01.TAXDATA
TAXDATA  VSAM INDX         4089  4096              1                    145 WPAV.WPAKMO.PAY01.TAXDATA
```

Because different devices have different blocking optimizations, research the correct block size for the device you are using. In general, for sequential files, half-track blocking optimizes I/O performance. Half-track blocking is a value in bytes of half the maximum value of the track, depending upon record format, device type, and record length.

VSAM Performance Improvement Opportunities

A possible opportunity for performance improvement of key-sequenced VSAM files with data and index clusters relates to file buffering. For a key-sequenced VSAM file that is directly accessed, buffering of the index component of the file may need to be increased. It is most efficient to assign to the number of buffers a value that is equal to the number of index set records plus the request parameter list's string number (RPL STRNO on the Data Set Characteristics report). The number of index set records for an index with two index levels is always one. By allocating more buffers, you place the entire index in storage. This placement reduces the number of physical I/Os (the EXCP COUNT on the Data Set Characteristics report) for the index component of the VSAM file and significantly reduces wait time.

As a general rule, if you always plan to access the file directly (DIR in the OPEN INTENT field on the Data Set Characteristics Supplement report, Figure 2-5 on page 2-6), you do not need to adjust the number of buffers to the data portion of the cluster. If, however, you change the application program so that this cluster is accessed sequentially (SEQ on the Data Set Characteristics Supplement report), increasing the number of the buffers to the data portion will reduce wait time. If the cluster is accessed dynamically (DYN on the

Data Set Characteristics Supplement report), allocating extra buffers to the cluster's index and data portions can improve performance.

If files experience control interval (CI) or control area (CA) splits, the section of the Data Set Characteristics Supplement that details %CA FREE and %CI FREE becomes relevant. CA splits (and, to a lesser extent, CI splits) incur wait. You can minimize CA splits by specifying FREESPACE when the cluster is defined, depending upon the function of the application.

Figure 2-5. Data Set Characteristics Supplement Report

```
                                ** DATA SET CHARACTERISTICS SUPPLEMENT **

DDNAME   ACCESS     DSNAME                                  OPEN INTENT\PROCESSING MODE
         METHOD

EMPLOYEE VSAM KSDS  WPAV.WPAKMO.PAY01.EMPLOYEE              OUTPUT,DIR
  FILE:  RECFM........                                VSAM:  FREESPACE... 916K    USER RECORDS...        9,999
         EXTENTS......  1                                    SHROPTS....(1,3)     LOGICAL OPERATIONS:
                                                             CI/CA......  150      DELETES........            0
                                                             %CI FREE.....  0      UPDATES........        6,937
                                                             %CA FREE.....  0      RETRIEVES......        6,938
                                                                                   INSERTS........            0

EMPLOYEE VSAM INDEX WPAV.WPAKMO.PAY01.EMPLOYEE              OUTPUT,DIR
  FILE:  RECFM........                                VSAM:  FREESPACE...  24K    USER RECORDS...            4
         EXTENTS......  1                                    SHROPTS....(1,3)     LOGICAL OPERATIONS:
                                                             CI/CA......   10      DELETES........            0
                                                             %CI FREE.....  0      UPDATES........            0
                                                             %CA FREE.....  0      RETRIEVES......            0
                                                             INDEX LVLS.    2      INSERTS........            0

PAYCHECK QSAM       WPAKMO.PAY01.OUTPUT.PAYCHEKS            OUTPUT
  FILE:  RECFM........FBM
         EXTENTS......  4


TAXDATA  VSAM KSDS  WPAV.WPAKMO.PAY01.TAXDATA              INPUT,DYN
  FILE:  RECFM........                                VSAM:  FREESPACE... 396K    USER RECORDS...           58
         EXTENTS......  1                                    SHROPTS....(1,3)     LOGICAL OPERATIONS:
                                                             CI/CA......  100      DELETES........            0
                                                             %CI FREE.....  0      UPDATES........            0
                                                             %CA FREE.....  0      RETRIEVES......            0
                                                                                   INSERTS........            0

TAXDATA  VSAM INDEX WPAV.WPAKMO.PAY01.TAXDATA              INPUT,DYN
  FILE:  RECFM........                                VSAM:  FREESPACE...  36K    USER RECORDS...            1
         EXTENTS......  1                                    SHROPTS....(1,3)     LOGICAL OPERATIONS:
                                                             CI/CA......   10      DELETES........            0
                                                             %CI FREE.....  0      UPDATES........            0
                                                             %CA FREE.....  0      RETRIEVES......            0
                                                             INDEX LVLS.    1      INSERTS........            0
```

If the VSAM file uses local shared resources (LSR), STROBE produces the VSAM LSR Pool Statistics report (Figure 2-6 on page 2-7). A directly accessed VSAM file that uses LSR and has efficient buffers performs fewer I/Os than a file with the same number of records that uses non-shared resources (NSR), thereby incurring less wait time. This efficiency occurs because the index component of files that take advantage of LSR can maintain sequence set records in storage, but an NSR file cannot. The VSAM LSR Pool Statistics report can help you evaluate the efficiency of the way LSR files are buffered by reporting the number of retrieves with I/O and the number of retrieves without I/O. If the number of retrieves with I/O is high compared to the number of retrieves without I/O, then investigating file buffering may offer an opportunity for performance improvement.

Figure 2-6. VSAM LSR Pool Statistics Report

```
                                        ** VSAM LSR POOL STATISTICS **

POOL  TYPE  STR KEY   BUF BUFNO   HBUFNO  --- RETRIEVES ---  ---- WRITES -----    ------------- HIPERSPACE -------------
  NO         NO LEN   LEN                   WITH    WITHOUT     USER   NONUSER        - READS -           - WRITES -
                                             I/O     I/O                          SUCCESSFUL FAILING  SUCCESSFUL FAILING
   1  DATA   37 255   512    30        0        0        0        0        0           0       0           0       0
   1  DATA   37 255  1024     4        0        0        0        0        0           0       0           0       0
   1  DATA   37 255  2048     3        0        0        0        0        0           0       0           0       0
   1  DATA   37 255  4096    70        0      908    30751       16        0           0       0           0       0
   1  DATA   37 255  8192    15        0       11       38        0        0           0       0           0       0
   1  DATA   37 255 12288    10        0        0        0        0        0           0       0           0       0
   1  DATA   37 255 24576     4        0        0        0        0        0           0       0           0       0
   1  INDEX     255   512    15        0        0        0        0        0           0       0           0       0
   1  INDEX     255  1024     5        0        0        0        0        0           0       0           0       0
   1  INDEX     255  2048    50        0        4    63189        0        0           0       0           0       0
   1  INDEX     255  4096    25        0        7      134        0        0           0       0           0       0
   1  INDEX     255  8192     5        0        0        0        0        0           0       0           0       0
   1  INDEX     255 12288     5        0        0        0        0        0           0       0           0       0
```

In addition to evaluating file characteristics on the Data Set Characteristics report and the Data Set Characteristics Supplement report, consider the possibility of internal contention.

Is Internal Contention Causing Wait Time?

Internal contention occurs if I/O is being performed on two files of the same step on the same device at the same time.

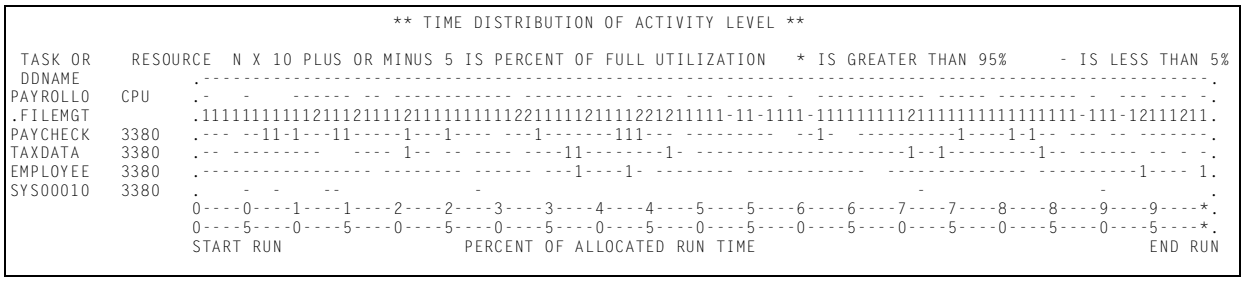
To check for internal contention, examine the I/O Facility Utilization Summary report, Figure 2-7 on page 2-7, to see if there are other files on the same device. This report lists the device and volume on which each data set resides and the amount of run time spent servicing each volume. If this report shows access to other files on the same device a high percentage of the time, refer to the Time Distribution of Activity Level report to see if the files are accessed at the same time.

Figure 2-7. I/O Facility Utilization Summary Report

```
                                        ** I/O FACILITY UTILIZATION SUMMARY **

 UNIT      DEVICE      CACHE  VOLUME DDNAME             RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:   .97%
  NO        TYPE       ELIG    ID                 I/O   SOLO     TOTAL        .00      1.00      2.00      3.00      4.00

  923  DA 3380K              WPA001 .FILEMGT             .87       .88        .********
  923  DA 3380K              WPA001 EMPLOYEE        I    .02       .02        .
  923  DA 3380K              WPA001 EMPLOYEE        O    .75       .75        .*******
  923  DA 3380K              WPA001 EMPLOYEE INDEX  I   1.18      1.18        .***********
  923  DA 3380K              WPA001 SYS00010             .07       .07        .
  923  DA 3380K              WPA001 TAXDATA         I    .65       .65        .******
  923  DA 3380K              WPA001 TAXDATA  INDEX  I    .76       .77        .*******
                                                        -----     -----
UNIT   923 TOTALS     C&DFW                            4.30      4.32

  92A  DA 3380K              WPA008 .FILEMGT             .01       .01        .
  92A  DA 3380K              WPA008 PAYCHECK        O   2.74      2.74        .**************************
                                                        -----     -----
UNIT   92A TOTALS     C&DFW                            2.75      2.75
```

The Time Distribution of Activity Level report (Figure 2-8 on page 2-8) shows, in vertical slices, the level of file access activity over time. Each vertical segment represents 1% of the measurement session. From this report, determine which tasks or ddnames are being accessed at times that overlap.

Figure 2-8. Time Distribution of Activity Level Report



If the Time Distribution of Activity Level report shows high access to multiple files concurrently, and the I/O Facility Utilization Summary report indicates that the files that are being accessed concurrently reside on the same volume, separating them may improve performance.

Another form of internal contention for a resource involves tape devices. If your application accesses a file on tape that spans multiple volumes, you will see the time associated with the tape mount and dismount attributed to .FILEMGT on the Resource Demand Distribution report. A period of total inactivity to the file in the Time Distribution of Activity Level report indicates this problem. Unless multiple tape units are allocated to service a multivolume sequential data set, access to a multivolume sequential data set will be held up during tape change.

External Contention

External contention occurs when more than one application is trying at the same time to access the same files or DASD volume. Although the Performance Profile does not specifically indicate external contention, the Resource Demand Distribution report may suggest this condition when the value in the SERVICED BY I/O column for a file is much smaller than the value in the CAUSING CPU WAIT column. If you suspect this condition, use the I/O Facility Utilization Summary report to determine which DASD your application programs are accessing, and then investigate applications that execute at the same time.

Are File Management Activities (.FILEMGT) Causing Wait Time?

Sometimes STROBE cannot determine the specific files that were responsible for wait, but can determine that it is related to system file management. .FILEMGT represents an aggregate of all file management activities such as file open and close, enqueues, tape mounts, catalog management, and similar overhead routines. STROBE can determine which module was waiting and where in your source code these overhead routines are invoked. You can then change your source code so that it invokes these overhead routines more economically.

If the Resource Demand Distribution report (Figure 2-3 on page 2-4) indicates that wait is related to .FILEMGT, refer to the Wait Time by Module report (Figure 2-9 on page 2-9) to identify the system services that are causing wait. The Wait Time by Module report tracks wait associated with .FILEMGT, ddnames, or tasks. Under MODULE NAME and SECTION NAME columns, this report identifies all the modules and control sections in which the target program was found to be waiting for CPU resources. A *control section* is the smallest unit of execution produced by a compiler or assembler that can be linked or replaced in a load module. Refer to the Attribution of CPU Wait Time report to identify the sites of invocation of service routines, as described in more detail below.

Figure 2-9. Wait Time by Module Report

```
                                        ** WAIT TIME BY MODULE **

MODULE   SECTION COMPRESSED   FUNCTION               RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:   .97%
 NAME      NAME    SECTION                            PAGE     TOTAL        .00     14.00     28.00     42.00     56.00

.IOCS    IGG019AR          QSAM PUT  NEXT BUFFER       .00      2.98        .++

.SVC     SVC 008           PROGRAM MANAGER/LOAD        .00       .01        .
.SVC     SVC 019           OPEN                        .00      4.00        .++
.SVC     SVC 020           CLOSE                       .00       .66        .
.SVC     SVC 026           CATALOG MANAGEMENT          .00     55.36        .++++++++++++++++++++++++++++++++++++++++++
.SVC     SVC 056           RESOURCE MANAGER/ENQUEUE    .00     20.52        .+++++++++++++++
.SVC     SVC 130           RACHECK                     .00       .23        .
                                                     -----     -----
.SVC     TOTALS            SUPERVISOR CONTROL          .00     80.78

.VSAM    IDA019L1          VSAM RECORD MANAGEMENT      .00      5.73        .++++
                                                     -----     -----
PROGRAM PAYROLLO TOTALS                                .00     89.49
```

If the Wait Time by Module identifies system service routines (such as SVCs reported under .SYSTEM) or language routines, check the Attribution of CPU Wait Time report (Figure 2-10 on page 2-10). This report identifies where selected service routines are invoked. For the system service routine identified in the header line, it shows under WAS INVOKED BY the hexadecimal offset from which the service routine was directly or indirectly invoked. If the report is indexed, it also shows the line number and the procedure name from which the system service routine was invoked. With this information, examine the source code to find ways to invoke the overhead routines so that they do not incur as much wait time.

Is Task Wait Causing Wait?

Task wait is all wait time not associated with file I/O or .FILEMGT. The Resource Demand Distribution report (Figure 2-3 on page 2-4) shows that tasks are causing wait when the RESOURCE column shows CPU as the type of resource used. Refer to the Wait Time By Module report, Figure 2-9 on page 2-9. This report lists all the modules, control sections, and operating system supervisory and service components in which the target program was found to be in the wait state.

If task wait is occurring in a system module, refer to the Attribution of CPU Wait Time report (Figure 2-10 on page 2-10) to identify program locations that called service routines. The Attribution of CPU Wait Time report shows the return address, and for indexed Performance Profiles, the lines of code and procedure names where the program invoked selected system service routines. Although you can rarely improve the performance of system service routines, you can examine the Attribution of CPU Wait Time report to see where they are invoked from and then restructure the code so that they are executed more economically. If subsystem routines are causing wait, refer to the STROBE subsystem Feature wait reports. These reports are explained in the appropriate STROBE Feature manual.

Figure 2-10. Attribution of CPU Wait Time Report

```
                                   ** ATTRIBUTION OF CPU WAIT TIME **

.SVC     SVC 026            CATALOG MANAGEMENT
--------------------------WAS INVOKED BY--------------------   -----------------VIA---------------------- WAIT TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                 PAGE  TOTAL

         PAYROLLO PAYROLLO   000E62    458      OPEN              IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE   .00   .01
         PAYROLLO PAYROLLO   001EF8    931      CLOSE             IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE   .00   .01
         TAXRTNO  TAXRTNO    000380    131      OPEN              IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE   .00 42.08
         TAXRTNO  TAXRTNO    0004BA    175      CLOSE             IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE   .00 17.05
                                                                                                               ----- -----
                                                                                                                .00 59.15
.SVC     SVC 056            RESOURCE MANAGER/ENQUEUE
--------------------------WAS INVOKED BY--------------------   -----------------VIA---------------------- WAIT TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                 PAGE  TOTAL

         PAYROLLO PAYROLLO   000E62    458      OPEN              IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE   .00   .01
         TAXRTNO  TAXRTNO    000380    131      OPEN              IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE   .00 17.66
         TAXRTNO  TAXRTNO    0004BA    175      CLOSE             IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE   .00  3.59
                                                                                                               ----- -----
                                                                                                                .00 21.26
```

What's Using CPU Time?

If your objective is to reduce CPU time and the Measurement Session Data report indicates that CPU time offers an opportunity for performance improvement, refer to the Program Section Usage Summary report (Figure 2-11 on page 2-10). Under the CPU TIME PERCENT TOTAL and SOLO columns, this report identifies which modules or control sections are the greatest consumers of this resource. The modules and control sections that use the greatest amount of CPU time are distinguished with a spike on the histogram to the right of the percentages. In addition, STROBE Feature reports offer further detail of CPU consumption by applications that use subsystems such as ADABAS/NATURAL, CICS, DB2, CA-IDMS, and IMS. For more information on STROBE Features, see the appropriate STROBE Feature manual.

Is Overhead Using Excessive CPU Time?

Overhead represents activities done on behalf of the application program by MVS system service routines, language routines, or subsystem routines. All of these are grouped under the label .SYSTEM on the Program Section Usage Summary report. On the Program Section Usage Summary report (Figure 2-11 on page 2-10) each type of system service has a detail line.

Figure 2-11. Program Section Usage Summary Report

```
                                   ** PROGRAM SECTION USAGE SUMMARY **

 MODULE  SECTION 16M   SECT FUNCTION                 CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  3.09%
   NAME  NAME    <,>   SIZE                            SOLO     TOTAL        .00     24.00     48.00     72.00     96.00

.SYSTEM  .COBLIB           COBOL LIBRARY SUBROUTINE   2.09      2.09        .
.SYSTEM  .IOCS             DATA MANAGEMENT SERVICES    .20       .20        .
.SYSTEM  .SVC              SUPERVISOR CONTROL        94.53     94.53        .****************************************
.SYSTEM  .VSAM             VIRTUAL STORAGE ACC METH   2.89      2.89        .*
                                                      -----     -----
.SYSTEM  TOTALS            SYSTEM SERVICES           99.71     99.71

PAYROLLO PAYROLLO <   14672                            .30       .30        .
                                                      -----     -----
PROGRAM PAYROLLO TOTALS                              100.00    100.00
```

When the Program Section Usage Summary shows high CPU time for .SYSTEM, distinguished by a spike on the histogram to the right of the entry, go to the Program Usage by Procedure report (Figure 2-12 on page 2-11) to identify further the users of this resource.

The first part of the Program Usage by Procedure report shows services reported under .SYSTEM. On this report, each control section of a system service has a detail line. STROBE identifies the function of the system services that the application program source code invoked, and the percent of CPU time used.

Figure 2-12. Program Usage by Procedure Report

```
                                        ** PROGRAM USAGE BY PROCEDURE **

          .SYSTEM     SYSTEM SERVICES                        .SVC      SUPERVISOR CONTROL
MODULE    SECTION        FUNCTION            INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  3.09%
  NAME      NAME                              LENGTH   SOLO     TOTAL        .00     12.00     24.00     36.00     48.00

SVC 019             OPEN                              12.64     12.64        .**********
SVC 020             CLOSE                             23.08     23.08        .********************
SVC 026             CATALOG MANAGEMENT                 9.15      9.15        .*******
SVC 056             RESOURCE MANAGER/ENQUEUE          47.76     47.76        .****************************************
SVC 060             ESTAE                               .20       .20        .
SVC 083             SMF                                1.39      1.39        .*
SVC 117             DEB VALIDITY CHECKING               .10       .10        .
SVC 130             RACHECK                             .20       .20        .
                                                      -----     -----
          .SVC      TOTALS                            94.52     94.52
```

For many system services that cause high CPU use, STROBE creates an Attribution of CPU Execution Time report (Figure 2-13 on page 2-11). This report identifies CPU execution time spent in certain service routines and identifies where they are invoked. STROBE attributes subsystem and language routines if your site has installed the STROBE Features that support them.

The Attribution of CPU Execution Time report pinpoints opportunities to avoid overhead by identifying the hexadecimal offsets and, if the Profile is indexed, the lines of code that invoke the system routines. Again, although normally you cannot improve the performance of system code itself, you may be able to restructure the invoking code so that it calls the overhead routines less frequently. Look at the invoking application code to see if the system service routines can be accessed more economically.

Figure 2-13. Attribution of CPU Execution Time Report

```
                                        ** ATTRIBUTION OF CPU EXECUTION TIME **

.SVC     SVC 019            OPEN
-------------------------WAS INVOKED BY---------------------   ------------------VIA----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME      MODULE   SECTION  FUNCTION                SOLO  TOTAL

         TAXRTNO  TAXRTNO    000380    131      OPEN             IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE 12.64 12.64
                                                                                                            ----- -----
                                                                                                           12.64 12.64
.SVC     SVC 020            CLOSE
-------------------------WAS INVOKED BY---------------------   ------------------VIA----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME      MODULE   SECTION  FUNCTION                SOLO  TOTAL

         TAXRTNO  TAXRTNO    0004BA    175      CLOSE            IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE 23.08 23.08
                                                                                                            ----- -----
                                                                                                           23.08 23.08
.SVC     SVC 026            CATALOG MANAGEMENT
-------------------------WAS INVOKED BY---------------------   ------------------VIA----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME      MODULE   SECTION  FUNCTION                SOLO  TOTAL

         TAXRTNO  TAXRTNO    000380    131      OPEN             IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE  6.77  6.77
         TAXRTNO  TAXRTNO    0004BA    175      CLOSE            IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE  2.39  2.39
                                                                                                            ----- -----
                                                                                                            9.15  9.15
.SVC     SVC 056            RESOURCE MANAGER/ENQUEUE
-------------------------WAS INVOKED BY---------------------   ------------------VIA----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME      MODULE   SECTION  FUNCTION                SOLO  TOTAL

         TAXRTNO  TAXRTNO    000380    131      OPEN             IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE 35.22 35.22
         TAXRTNO  TAXRTNO    0004BA    175      CLOSE            IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE 12.54 12.54
                                                                                                            ----- -----
                                                                                                           47.76 47.76
```

Is the Source Code Causing Excessive CPU Time Consumption?

On the Program Section Usage Summary report (Figure 2-11 on page 2-10) each control section of an application has a detail line. If the Program Section Usage Summary report showed that application code rather than overhead routines was responsible for CPU consumption, read the Program Usage by Procedure report (Figure 2-12 on page 2-11) for the indicated user program. The Program Usage by Procedure report displays which procedure names and lines of code were using CPU.

On the Program Usage by Procedure report, each codeblock has a detail line. A *codeblock* is a division of a control section whose size in bytes equals the specified report resolution. The detail lines on the Program Usage by Procedure report relate CPU use to hexadecimal offsets within the program and, for an indexed Performance Profile, to source code lines and procedure names. Again, spikes in the histogram indicate the significant users. (For more information about indexed Performance Profiles, see "Indexed Performance Profiles" on page 2-26.)

Note: If you know that you are looking for a reduction in CPU time, instead of reviewing the Program Section Usage Summary report and the Program Usage by Procedure report, go directly to the Most Intensively Executed Procedure report (Figure 2-14 on page 2-12). This report shows you the ten procedures that used the most CPU time, ranked in order of their CPU consumption. Refer to this report to determine which procedures executing in the measured job step used the most CPU time.

Figure 2-14. Most Intensively Executed Procedures Report

** MOST INTENSIVELY EXECUTED PROCEDURES **

MODULE NAME	SECTION NAME	LINE NUMBER	PROCEDURE/FUNCTION NAME	STARTING LOCATION	PROCEDURE LENGTH	CPU TIME PERCENT SOLO	CPU TIME PERCENT TOTAL	CUMULATIVE PERCENT SOLO	CUMULATIVE PERCENT TOTAL
STRBSAM1	SAMPLEW	70	WAY1-SUBSCRIPT	00096A	94	32.64	32.64	32.64	32.64
STRBSAM1	SAMPLEW	76	WAY2-INDEXED	0009C8	76	29.75	29.75	62.39	62.39
STRBSAM1	SAMPLEW	82	WAY3-ABS-SUB	000A14	96	22.31	22.31	84.70	84.70
STRBSAM1	SAMPLEW	89	WAY4-STRAIGHT	000A74	36	4.55	4.55	89.25	89.25
.SVC	SVC 019		OPEN			1.65	1.65	90.90	90.90
STRBSAM1	SAMPLEW	95	END-JOB	000A98	364	1.24	1.24	92.14	92.14
STRBSAM1	SAMPLEW	59	MAIN-3	00088E	220	1.24	1.24	93.38	93.38
STRBSAM1	SAMPLEW	54	MAIN-LINE	0007CE	192	1.24	1.24	94.62	94.62
.SVC	SVC 013		TERMINATION			1.24	1.24	95.86	95.86
.IOCS	IECOSCR1		PHYSICAL IOCS			1.24	1.24	97.10	97.10

Incorporate Changes

STROBE can pinpoint the location of inefficiencies within the source code. After you interpret the Performance Profile, decide where to economize on the resources that your application uses. Examine the application, make corrections, test them, and measure again with STROBE. Remeasuring enables you both to see the results of the changes and to identify other possible opportunities for improvement that may have been masked by the first performance problem.

Summary of Reports to Analyze

The Performance Profile reports, shown next, reflect the interpretation explained in "STROBE Performance Profile Interpretation Steps" on page 2-1.

The following reports are related to wait time:

Report Name	Type of Information
Resource Demand Distribution	Identifies what is causing wait
Data Set Characteristics	Identifies files, buffers, blocksize, etc.

Report Name	Type of Information
I/O Facility Utilization Summary	Identifies run time by ddname, volume, and unit
DASD Usage by Cylinder	Identifies run time by cylinder, ddname, and volume
Data Set Characteristics Supplement	Identifies intent, logical operations, etc.
VSAM LSR Pool Statistics	Provides resource information for data sets allocated to local shared resources
Time Distribution of Activity Level	Gives time line of resource activity
Attribution of CPU Wait Time	Identifies wait caused by callers of service routines
Wait Time by Module	Identifies in which module the job is waiting
Subsystem-specific reports	Provide CPU wait time information

The following reports are related to CPU time:

Report Name	Type of Information
Program Section Usage Summary	Identifies CPU consumption at module level or by type of service routine
Program Usage by Procedure	Breaks down CPU consumption within modules or specific service routine
Attribution of CPU Execution Time	Identify CPU used by the callers of service routines
Most Intensively Executed Procedures	Identifies the top ten CPU consumers
Subsystem-specific reports	Provide CPU consumption information

The following reports provide additional reference information:

Report Name	Type of Information
Coupling Facility Activity report	Identifies system-wide coupling facility activity
Token - Cross Reference report	Reconciles all tokens with their original long names

Optional STROBE Features generate reports that provide additional support for specific operating environments such as ADABAS/NATURAL, CA-IDMS, CICS, CSP, DB2, COOL:Gen, MQSeries, Java, UNIX System Services, and IMS. See the appropriate STROBE Feature manual for more information.

Performance Profile Interpretation: A Case Study

This section provides a case study of how following the STROBE Performance Profile Interpretation flowchart can lead to improvements in application performance. To

compute and print paychecks, the program (PAYROLL0) analyzed in this case study accesses files that contain employee record information, time sheet data, payroll information, and state tax rates. Although the code did not change, the run time recently increased dramatically. The Personnel office is concerned that paychecks may arrive late. An application developer measured the program and produced a Performance Profile. Excerpts are shown below.

The STROBE Performance Profile leads you to the lines of code that are responsible for the increased run time. This case study demonstrates three opportunities for performance improvement. The section following the case study outlines possible opportunities for improvement not covered in the case study. Appendix B, "Example Program and STROBE Performance Profile" contains the source code for the case study program and the complete Performance Profile.

Valid Performance Profile?

The Performance Profile that results from measuring the example program is called PAYROLL0. The first step in evaluating this Performance Profile is to examine the Measurement Session Data report (Figure 2-15 on page 2-14) to determine that the reports in the Profile reflect measurement of the intended job step with the intended parameters, and that the information is statistically valid. With PAYROLL0, verify that these fields contain the correct or expected information.

Figure 2-15. Measurement Session Data Report (PAYROLL0)

```
                                        ** MEASUREMENT SESSION DATA **

------- JOB ENVIRONMENT --------        ------- MEASUREMENT PARAMETERS ------        ----- MEASUREMENT STATISTICS ----

PROGRAM MEASURED   -    PAYROLL0       ESTIMATED SESSION TIME  -      45 MIN       CPS TIME PERCENT -          10.51
JOB NAME           -    WPAJEAG0       TARGET SAMPLE SIZE      -      10,000       WAIT TIME PERCENT -         89.49
JOB NUMBER         -    JOB01541       REQUEST NUMBER   (Q)    -          36       RUN MARGIN OF ERROR PCT  -    .97
STEP NAME   -             RUNAPP                                                  CPU MARGIN OF ERROR PCT  -   2.99

DATE OF SESSION    -  04/12/1999       SYS REQ - ALL SYSTEMS                       TOTAL SAMPLES TAKEN -      10,248
TIME OF SESSION    -    11:25:53                                                  TOTAL SAMPLES PROCESSED -  10,248
CONDITION CODE     -      C-0000       MODULE MAPPING BASELINE -           2       INITIAL SAMPLING RATE-   3.70/SEC
                                       BASELINE OVERRIDE -            TAXRTNO     FINAL SAMPLING RATE  -   3.70/SEC
SYSTEM -             ESA SP5.2.0       LIBRARY -             WPAKMO.USER.LOAD
DFSMS              -       1.2.0                                                 SESSION TIME -   52 MIN  5.98 SEC
CPU MODEL          -   3090-600S                                                 CPU TIME -        5 MIN 35.50 SEC
SMF/SYSTEM ID -       MVST/TLP01       --------- REPORT PARAMETERS ---------     WAIT TIME    -   45 MIN 53.92 SEC
                                                                                 STRETCH TIME -    0 MIN 36.56 SEC
REGION SIZE BELOW 16M  -  4,160K       REPORT RESOLUTION    -       64 BYTES
REGION SIZE ABOVE  -     32,768K       SORTSIZE             -      999,999       SRB TIME     -    0 MIN 17.40 SEC
                                       LINES/PAGE           -           60       SERVICE UNITS-             334494
PTF LVL- 2.3.0.FS000000/FS000000
                                       DASD= 2.0%  DASDGAP= 5                    PAGES IN-      0     OUT-       0
                                                                                 PAGING RATE  -           0.00/SEC
                                       DATE FORMAT               MM/DD/YYYY      EXCPS -     38,093      12.19/SEC
                                       TIME FORMAT (24 HOURS)      HH:MM:SS

SAMPLE DATA SET  - STROBE.WPAJEAG0.S001D001
```

In the JOB ENVIRONMENT section verify:

- PROGRAM MEASURED is PAYROLL0.
- JOB NAME is WPAJEAG0.
- JOB NUMBER is JOB 01541.
- STEP NAME is RUNAPP.
- CONDITION CODE, which is the return code, is C-0000. PAYROLL0 ran to completion and returned a return code of 0 to the operating system.
- SYSTEM and CPU MODEL are System ESA SP5.2.0 and model 3090-600S, the version of MVS and the machine model that normally support production jobs.
- SMF/SYSTEM ID is MVST/TLP01, the system on which production jobs commonly run.

In the MEASUREMENT PARAMETERS section verify:

- ESTIMATED SESSION TIME is 45 minutes (this is an estimate of the program's run time).
- TARGET SAMPLE SIZE is 10,000.
- REQUEST NUMBER (Q) indicates that the number of the request is 36 and the "Q" indicates that it was a queued request.

In the MEASUREMENT STATISTICS section verify:

- RUN MARGIN OF ERROR PCT, the margin of error for the percentages in the reports that detail run time, is less than or equal to 2.00, indicating that the information in the Performance Profile is statistically valid.
- The CPU MARGIN OF ERROR PCT is 2.99% which, since it is under the ceiling value of 10, indicates that enough samples were taken to provide valid information in the reports that detail CPU time.

CPU or Wait?

Note the following:

- SESSION TIME, the amount of time the session ran, is 52 minutes and 5.98 seconds.
- CPS TIME PERCENT is 10.51 and the WAIT TIME PERCENT is 89.49.
- CPU TIME, the amount of CPU resources STROBE observed the application using, is 5 minutes and 35.50 seconds.
- WAIT TIME, the amount of time STROBE observed the application waiting to use CPU resources, is 45 minutes and 53.92 seconds.

The actual session time of 52 minutes and 5.98 seconds is slightly longer than the estimated session time of 45 minutes. The majority of this time, 45 minutes and 53.92 seconds, is wait time. Because the application experiences a significant amount of wait time, it merits further investigation.

Knowing that you would like to reduce wait time, refer to Figure 2-1 on page 2-2 to determine which report to examine.

What's Causing Wait?

The flowchart, Figure 2-1 on page 2-2, indicates that to determine the cause of wait the next report to examine is the Resource Demand Distribution report.

Identifying Demand for Resources

On the Resource Demand Distribution report shown in Figure 2-16 on page 2-16, .FILEMGT has the largest value in the CAUSING CPU WAIT column under PERCENT OF RUN TIME SPENT. This value is 80.47 %. STROBE groups under .FILEMGT all file processing activity that it does not relate to a specific ddname. In this situation, the Performance Profile indicates the code that is responsible for invoking the overhead routines. For a high percentage identified in .FILEMGT, the flowchart directs you to the Wait Time By Module report.

Figure 2-16. Resource Demand Distribution Report (PAYROLL0)

```
                                        ** RESOURCE DEMAND DISTRIBUTION **

                      ---- PERCENT OF RUN TIME -----     ----- PERCENT OF RUN TIME SPENT -------   CUMULATIVE PERCENTAGES
TASK OR     RESOURCE     SERVICED  SERVICED  SERVICED      SOLO      SOLO      SOLO     CAUSING         SOLO      CAUSING
 DDNAME                   BY CPU    BY I/O  BY EITHER     IN CPU    IN I/O  IN EITHER   CPU WAIT         TIME    CPU WAIT

PAYROLL0    CPU            1.22       .00       1.22       1.21       .00       1.21        .24         1.21        .24
.FILEMGT                   8.58       .89       9.46       8.57       .88       9.46      80.47        10.67      80.71
PAYCHECK    3380            .00      2.74       2.74        .00      2.74       2.74       2.98        13.41      83.69
TAXDATA     3380            .67      1.42       2.10        .67      1.41       2.09       2.40        15.50      86.09
EMPLOYEE    3380            .04      1.95       1.99        .04      1.95       1.99       3.33        17.49      89.42
SYS00010    3380            .00       .07        .07        .00       .07        .07        .07        17.56      89.49
```

Identifying Modules Experiencing Wait

As shown in Figure 2-17 on page 2-16, the Wait Time by Module report lists the modules that STROBE found waiting for CPU resources. It shows that in module SVC 026 and in module SVC 056 the application was waiting a significant amount of time. The application was waiting in module SVC 026 55.36 % of the time and in module SVC 056 20.52 % of the time. The concentration of wait in these two routines tells you that in these supervisor call (SVC) routines, which are MVS system service routines, the application experiences the majority of wait time. Refer to the Attribution of CPU Wait Time report (Figure 2-18 on page 2-17) to find the lines of source code that invoked these overhead routines.

Figure 2-17. Wait Time By Module Report (PAYROLL0)

```
                                                ** WAIT TIME BY MODULE **

MODULE   SECTION COMPRESSED   FUNCTION                RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:   .97%
 NAME      NAME    SECTION                              PAGE     TOTAL        .00     14.00     28.00     42.00     56.00

.IOCS    IGG019AR          QSAM PUT  NEXT BUFFER        .00      2.98        .++

.SVC     SVC 008           PROGRAM MANAGER/LOAD         .00       .01        .
.SVC     SVC 019           OPEN                         .00      4.00        .++
.SVC     SVC 020           CLOSE                        .00       .66        .
.SVC     SVC 026           CATALOG MANAGEMENT           .00     55.36        .++++++++++++++++++++++++++++++++++++++++++
.SVC     SVC 056           RESOURCE MANAGER/ENQUEUE     .00     20.52        .+++++++++++++++
.SVC     SVC 130           RACHECK                      .00       .23        .
                                                       -----     -----
.SVC     TOTALS            SUPERVISOR CONTROL           .00     80.78

.VSAM    IDA019L1          VSAM RECORD MANAGEMENT       .00      5.73        .++++
                                                       -----     -----
PROGRAM PAYROLL0 TOTALS                                 .00     89.49
```

Identifying Callers of Waiting Modules

The Attribution of CPU Wait Time report shows that SVC 026 was invoked from both PAYROLL0 and TAXRTN0. It was invoked by PAYROLL0 at line 458, which is an OPEN procedure. SVC 026 was also invoked by TAXRTN0 at line 131, which is an OPEN procedure, and line 175, which is a CLOSE procedure. Similarly, SVC 056 was invoked by both PAYROLL0 and TAXRTN0 and is called from the same locations. The percentage of wait time for TAXRTN0 under WAIT TIME % TOTAL column for the invocation of both service routines is high (55.33% and 17.17% respectively). Because a high proportion of the wait for this application is caused by the execution of lines 131 and 175 in TAXRTN0, these lines of source code should be investigated.

Figure 2-18. Attribution of CPU Wait Time (PAYROLL0)

```
                                  ** ATTRIBUTION OF CPU WAIT TIME **


.SVC     SVC 026            CATALOG MANAGEMENT
--------------------------WAS INVOKED BY---------------------   -----------------VIA---------------------- WAIT TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                 PAGE  TOTAL

         .COBLIB  IGZCPCO  IGZEVOP    VSAM OPEN                  SVC 019           OPEN                      .00   .01
         PAYROLLO PAYROLLO   000E6A    458      OPEN              IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE  .00   .02
         TAXRTN0  TAXRTN0    000380    131      OPEN              IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE  .00 34.71
         TAXRTN0  TAXRTN0    0004BA    175      CLOSE             IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE  .00 20.62
                                                                                                              ----- -----
                                                                                                               .00 55.36

.SVC     SVC 056            RESOURCE MANAGER/ENQUEUE
--------------------------WAS INVOKED BY---------------------   -----------------VIA---------------------- WAIT TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                 PAGE  TOTAL

         .COBLIB  IGZCPCO  IGZEVOP    VSAM OPEN                  SVC 019           OPEN                      .00   .01
         TAXRTN0  TAXRTN0    000380    131      OPEN              IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE  .00 17.17
         TAXRTN0  TAXRTN0    0004BA    175      CLOSE             IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE  .00  3.34
                                                                                                              ----- -----
                                                                                                                .00 20.5
```

Examining the Source Code

The next step is to examine the related COBOL source code. When the payroll application needs to get a state income tax percentage rate for an employee, subroutine TAXRTN0:

- opens the TAXDATA file
- gets the tax rate for the specific state
- closes the TAXDATA file

The increase in wait time is incurred because the subroutine TAXRTN0 is opening and closing the TAXDATA file for each call. This repetition occurs because Connecticut recently initiated an income tax and many of this company's employees live in Connecticut. Subsequently, the number of times the TAXDATA file was opened and closed increased dramatically, incurring an unacceptable amount of wait time.

Improving the Source Code

The correction to this problem is to invoke TAXRTN0 so as to minimize the number of COBOL open and close statements. You could change the code to open the file once at the beginning of the program and close it at the end. In this case, the file was small enough to build a table in working storage in which the rates from the tax table file TAXDATA are stored. This method enables you to incur only one open and close and reduce the number of I/Os. Before correcting the code, however, look at the CPU opportunity that this Performance Profile presents.

What's Using CPU Time?

Although you have determined why the application's run time is increased, the Personnel office also commented that the application was also expensive to run, which relates directly to the application's use of CPU time. Since the Performance Profile is valid, check to see if there is an opportunity to reduce CPU time. You may be able to reduce the 5 minutes and 35.50 seconds of CPU time shown on the Measurement Session Data report. To see what module is using CPU time, refer to the Program Section Usage Summary, Figure 2-19 on page 2-18, as the STROBE Performance Profile Interpretation flowchart (Figure 2-1 on page 2-2) indicates.

Identifying CPU Time Consumption by Program Section

The Program Section Usage Summary report, Figure 2-19 on page 2-18, identifies the amount of CPU time used by modules and control sections. The module PAYROLL0,

which you are responsible for coding, used very little of the CPU time. STROBE assigns most of the use of CPU time to the module .SYSTEM and to a section named .SVC. STROBE groups all system services under .SYSTEM. The section .SVC within .SYSTEM used 90.71% of CPU time. Note that this section is distinguished by a large spike on the histogram. To further investigate CPU consumption, refer to the Program Usage by Procedure report for .SYSTEM .SVC, Figure 2-20 on page 2-18.

Figure 2-19. Program Section Usage Summary Report (PAYROLL0)

```
                                        ** PROGRAM SECTION USAGE SUMMARY **

 MODULE  SECTION 16M   SECT FUNCTION                 CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.99%
   NAME  NAME    <,>   SIZE                           SOLO     TOTAL         .00     23.00     46.00     69.00     92.00

.SYSTEM  .COBLIB            COBOL LIBRARY SUBROUTINE   1.95      1.95        .
.SYSTEM  .IOCS              DATA MANAGEMENT SERVICES    .19       .19        .
.SYSTEM  .SVC               SUPERVISOR CONTROL        90.53     90.71        .****************************************
.SYSTEM  .VSAM              VIRTUAL STORAGE ACC METH   6.78      6.78        .**
                                                      -----     -----
.SYSTEM  TOTALS             SYSTEM SERVICES           99.45     99.63

PAYROLL0 PAYROLL0 <   14680                            .37       .37        .
                                                      -----     -----
PROGRAM PAYROLL0 TOTALS                               99.81    100.00
```

Identifying CPU Time Consumption by Procedures

The Program Usage by Procedure report (Figure 2-20 on page 2-18) shows that SVC 019 used 22.19% of the CPU time, SVC 020 used 16.53% of CPU time, SVC 026 used 8.54% of CPU time, and SVC 056 used 34.35% of CPU time.

Figure 2-20. Program Usage by Procedure Report (PAYROLL0)

```
                                        ** PROGRAM USAGE BY PROCEDURE **

         .SYSTEM     SYSTEM SERVICES                    .SVC      SUPERVISOR CONTROL

MODULE   SECTION        FUNCTION            INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.99%
  NAME      NAME                              LENGTH   SOLO     TOTAL         .00      9.00     18.00     27.00     36.00

SVC 019           OPEN                                22.10     22.19        .************************
SVC 020           CLOSE                               16.53     16.53        .******************
SVC 026           CATALOG MANAGEMENT                   8.54      8.54        .*********
SVC 040           TASK MANAGER/EXTRACT                  .09       .09        .
SVC 048           RESOURCE MANAGER/DEQUEUE              .56       .56        .
SVC 056           RESOURCE MANAGER/ENQUEUE            34.35     34.35        .************************************
SVC 060           ESTAE                                 .93       .93        .*
SVC 083           SMF                                  1.76      1.76        .*
SVC 117           DEB VALIDITY CHECKING                 .09       .09        .
SVC 130           RACHECK                              5.57      5.66        .******
                                                      -----     -----
         .SVC      TOTALS                            90.52     90.70
```

With this information, refer to the Attribution of CPU Execution Time report (Figure 2-21 on page 2-19) to see what lines of application code are responsible for invoking these SVCs.

Figure 2-21. Attribution of CPU Execution Time Report (PAYROLL0)

```
                                   ** ATTRIBUTION OF CPU EXECUTION TIME **

.SVC     SVC 019            OPEN
-------------------------WAS INVOKED BY---------------------   -----------------VIA----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                SOLO  TOTAL

         TAXRTN0  TAXRTN0    000380    131      OPEN              IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE 22.10 22.19
                                                                                                             ----- -----
                                                                                                            22.10 22.19

.SVC     SVC 020            CLOSE
-------------------------WAS INVOKED BY---------------------   -----------------VIA----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                SOLO  TOTAL

         PAYROLL0 PAYROLL0   001F00    931      CLOSE             IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE   .09   .09
         TAXRTN0  TAXRTN0    0004BA    175      CLOSE             IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE 16.43 16.43
                                                                                                             ----- -----
                                                                                                            16.53 16.53

.SVC     SVC 026            CATALOG MANAGEMENT
-------------------------WAS INVOKED BY---------------------   -----------------VIA----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                SOLO  TOTAL

         TAXRTN0  TAXRTN0    000380    131      OPEN              IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE  6.31  6.31
         TAXRTN0  TAXRTN0    0004BA    175      CLOSE             IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE  2.23  2.23
                                                                                                             ----- -----
                                                                                                             8.54  8.54

.SVC     SVC 056            RESOURCE MANAGER/ENQUEUE
-------------------------WAS INVOKED BY---------------------   -----------------VIA----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                SOLO  TOTAL

         TAXRTN0  TAXRTN0    000380    131      OPEN              IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE 26.37 26.37
         TAXRTN0  TAXRTN0    0004BA    175      CLOSE             IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE  7.99  7.99
                                                                                                             ----- -----
                                                                                                            34.35 34.35
```

Attributing CPU Execution Time

The Attribution of CPU Execution Time report, Figure 2-21 on page 2-19, shows that SVC 019 was invoked by TAXRTN0 at line 131, which is an OPEN procedure. SVC 020 was invoked by TAXRTN0 at line 175, which is a CLOSE procedure. Similarly, SVC 020, SVC 026, and SVC 056 were invoked by lines 131 and 175 of TAXRTN0. (SVC 020 was also invoked at line 931 of the program PAYROLL0, which is a CLOSE procedure, but because invocation from this line accounts for only .09% of CPU time, it does not offer an opportunity to improve performance.)

Note that the Attribution of CPU Execution Time report is indexed; for the modules identified, it displays the source code line number and the procedure name in addition to the hexadecimal starting location and interval length. All indexed reports that show source code provide this information. (For more information about indexed Performance Profiles, see “Indexed Performance Profiles” on page 2-26.)

Identifying the Source Code

STROBE points to the same areas of source code, lines 131 and 175, as those you found when you followed the opportunity to reduce wait time. STROBE enables you to pinpoint those critical path lines of code that caused the application suddenly to become inefficient. It points to the COBOL OPEN and CLOSE statements where the program PAYROLL0 calls the external subroutine TAXRTN0 each time the program needs to retrieve a state tax rate. This application worked efficiently when only a few of the workers lived in states that imposed an income tax. But in the mid 1990s, Connecticut initiated an income tax. Because most of the employees reside in Connecticut, the section of the program that retrieved the state tax rate was no longer efficient.

Changing the Source Code

The next step is to modify the source code. One solution is to build a table in working storage that contains data loaded from the state tax rate information file, opening and closing the file just once. With this change, program PAYROLL0 makes one call to the routine that loads the tax data, rather than invoking it repeatedly.

Identifying CPU Consumption Related to Source Code

In this case study, the Program Section Usage Summary, Figure 2-19 on page 2-18, shows that module PAYROLL0 and section name PAYROLL0 used only a total of .37% of the CPU time. STROBE identifies the lines of code responsible for this CPU consumption. The indexed Program Usage by Procedure report, Figure 2-22 on page 2-20, identifies the line numbers, procedure names, and hexadecimal offsets for the responsible lines of code. For example, line 837 is an ADD statement that used .28% of the CPU time.

Because this example does not show significant amounts of CPU resource use, you probably would not pursue the performance analysis here.

Figure 2-22. Program Usage By Procedure Report for Module (PAYROLL0)

```
                                    ** PROGRAM USAGE BY PROCEDURE **

MODULE - PAYROLL0                      SECTION - PAYROLL0
                                       SOURCE LANGUAGE - IBM COBOL II                                    VC40X26

  LINE      PROCEDURE          STARTING      INTERVAL  CPU TIME PERCENT      CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.99%
 NUMBER       NAME             LOCATION       LENGTH    SOLO     TOTAL        .00       .50      1.00      1.50      2.00

         DATA DIVISION          000000          6566     .00       .00      - .
    837     ADD                 0019A6            82     .28       .28        .*****
    842  0640-EXIT              0019F8           156     .00       .00      - .
    865  0700-WRIT..HECK-FILE   001A94            94     .09       .09        .*
    872     WRITE               001AF2          7782     .00       .00      - .
                                                       -----     -----
 SECTION  PAYROLL0  TOTALS                              .37       .37
```

Measure Again with STROBE

After making the appropriate changes to the code, measure the modified program, PAYROLL1, and evaluate the resulting Performance Profile. This Performance Profile shows the effects of your changes and may reveal opportunities to improve performance that were not apparent the first time you measured the program. You may find opportunities to reduce wait time even further.

The change to the application code dramatically reduced wait time. On the Measurement Session Data report (Figure 2-23 on page 2-21) the wait time for PAYROLL1 is 10 minutes and 2.35 seconds, which is a reduction of 35 minutes and 51.57 seconds. Note also that CPU time is 14.18 seconds, which is a reduction of 5 minutes and 21.32 seconds.

The WAIT TIME PERCENT value can increase even when the actual wait time experienced has decreased. For example, the WAIT TIME has decreased from 45 minutes and 35.50 seconds in PAYROLL0 to 10 minutes and 58.58 seconds in PAYROLL1. The WAIT TIME PERCENT, however, has increased from 89.49 in PAYROLL0 to 99.49 in PAYROLL1. This increase occurs because in PAYROLL1 the percentage is a larger portion of a much smaller number.

The Measurement Session Data report for PAYROLL1 shows that there may be additional performance improvement opportunities. The WAIT TIME PERCENT field shows that 99.49% of the time the application is in a wait state. Because the percentage of wait time is high, it merits investigation.

Figure 2-23. Measurement Session Data Report (PAYROLL1)

```
                                   ** MEASUREMENT SESSION DATA **

-------- JOB ENVIRONMENT --------        ------- MEASUREMENT PARAMETERS ------        ----- MEASUREMENT STATISTICS ----

 PROGRAM MEASURED   -    PAYROLL1        ESTIMATED SESSION TIME  -       1 MIN        CPS TIME PERCENT -           0.51
 JOB NAME           -    WPAJEAG1        TARGET SAMPLE SIZE      -      10,000        WAIT TIME PERCENT -         99.49
 JOB NUMBER         -    JOB06728        REQUEST NUMBER   (Q)    -        254        RUN MARGIN OF ERROR PCT  -    .76
 STEP NAME   -             RUNAPP                                                    CPU MARGIN OF ERROR PCT  -  10.57

 DATE OF SESSION    -  05/18/1999        SYS REQ -  TLP01                           TOTAL SAMPLES TAKEN -      16,848
 TIME OF SESSION    -    11:11:17                                                   TOTAL SAMPLES PROCESSED -  16,848
 CONDITION CODE     -      C-0000        MODULE MAPPING BASELINE -          2       INITIAL SAMPLING RATE- 166.67/SEC
                                         BASELINE OVERRIDE -           TAXRTN1     FINAL SAMPLING RATE  - 166.67/SEC
 SYSTEM -             ESA SP5.2.0
 DFSMS              -       1.2.0                                                  SESSION TIME -   10 MIN 58.58 SEC
 CPU MODEL          -   3090-600S        --------- REPORT PARAMETERS ---------     CPU TIME -        0 MIN 14.18 SEC
 SMF/SYSTEM ID -       MVST/TLP01                                                  WAIT TIME    -   10 MIN  2.35 SEC
                                         REPORT RESOLUTION    -       64 BYTES     STRETCH TIME -    0 MIN 42.05 SEC
 REGION SIZE BELOW 16M  -  4,160K        SORTSIZE             -      999,999
 REGION SIZE ABOVE  -     32,768K        LINES/PAGE           -           60       SRB TIME     -    0 MIN  8.34 SEC
                                                                                  SERVICE UNITS-              14889
 PTF LVL- 2.3.0.FS000000/FS000000        DASD= 2.0%  DASDGAP= 5
                                                                                  PAGES IN-      3     OUT-       0
                                         DATE FORMAT               MM/DD/YYYY     PAGING RATE  -           0.00/SEC
                                         TIME FORMAT (24 HOURS)       HH:MM:SS    EXCPS -     37,423      56.82/SEC


 SAMPLE DATA SET  - WPAJEA.V1.WPAJEAG1.S001D001
```

Reviewing the Resource Demand Distribution Report

To investigate wait time further, refer to the Resource Demand Distribution report (Figure 2-24 on page 2-21). In the Resource Demand Distribution report the PERCENT OF RUN TIME SPENT in the CAUSING CPU WAIT column shows that ddnames EMPLOYEE and PAYCHECK are experiencing a high degree of wait time. .FILEMGT has a very small amount (.84%) so you know that file processing is not causing the wait. The ddname EMPLOYEE, which contains personal information about employees, is causing CPU to wait 55.05% of the time. PAYCHECK, which contains employee payroll information, is causing CPU to wait 43.28% of the time. As the Performance Profile Interpretation Flowchart indicates, refer to the Data Set Characteristics report (Figure 2-25 on page 2-22) to see how the files are defined.

Figure 2-24. Resource Demand Distribution Report (PAYROLL1)

```
                                   ** RESOURCE DEMAND DISTRIBUTION **

                      ---- PERCENT OF RUN TIME -----    ----- PERCENT OF RUN TIME SPENT -------   CUMULATIVE PERCENTAGES
TASK OR    RESOURCE     SERVICED   SERVICED  SERVICED      SOLO      SOLO      SOLO     CAUSING        SOLO      CAUSING
 DDNAME                  BY CPU    BY I/O  BY EITHER     IN CPU    IN I/O  IN EITHER   CPU WAIT        TIME    CPU WAIT

PAYROLL1   CPU              .18       .00       .18        .18       .00       .18        .11         .18         .11
PAYCHECK   3380             .01     40.69     40.69        .01     40.69     40.69      43.28       40.87       43.39
EMPLOYEE   3380             .02     32.89     32.91        .02     32.87     32.89      55.05       73.76       98.44
.FILEMGT                    .30       .18       .49        .30       .15       .46        .84       74.22       99.28
TIMSHEET   3380             .01       .08       .09        .01       .08       .09        .17       74.31       99.45
STEPLIB    3380             .00       .04       .04        .00       .04       .04        .04       74.35       99.49
TAXDATA    3380             .00       .01       .01        .00       .01       .01        .01       74.36       99.50
```

Reviewing the Data Set Characteristics Report

The Data Set Characteristics report, Figure 2-25 on page 2-22, shows the characteristics of the two files in which you are interested. Note that PAYCHECK is a QSAM file. EMPLOYEE is a VSAM key sequenced data set with a data component (identified by KSDS in the ACCESS METHOD column) and an index component (identified by INDX in the ACCESS METHOD column).

Review the information on the Data Set Characteristics report. Notice the following:

- The block size or CI size (BLK/CI SIZE).
- The number of blocks transferred (EXCP COUNTS).

- The number of buffers (BUF NO) for the data and index portions for VSAM.
- The POOL NO field is blank, indicating that the VSAM file uses non-shared resources.

Figure 2-25. Data Set Characteristics Report (PAYROLL1)

```
                                        ** DATA SET CHARACTERISTICS **

DDNAME     ACCESS    POOL  REC  BLK/CI    HBUF  BUF  RPL  -SPLITS-     EXCP DATA SET NAME
           METHOD      NO  SIZE  SIZE       NO   NO STRNO CI    CA   COUNTS

EMPLOYEE VSAM KSDS          150  4096             2   1               7264 WPAV.WPAJEA.PAY01.EMPLOYEE
EMPLOYEE VSAM INDX         4089  4096             1                  13997 WPAV.WPAJEA.PAY01.EMPLOYEE
PAYCHECK QSAM               132  1320             5                  16100 WPAJEA.PAY01.OUTPUT.PAYCHEKS
STEPLIB  BPAM                    8906                                    5 WPAJEA.USER.LOAD
TAXDATA  VSAM KSDS           10  4096             2   1                  WPAV.WPAJEA.PAY01.TAXDATA
TAXDATA  VSAM INDX         4089  4096             1                      WPAV.WPAJEA.PAY01.TAXDATA
TIMSHEET VSAM ESDS           35  4096             2   1                 57 WPAV.WPAJEA.PAY01.TIMSHEET
```

QSAM Blocking and Buffering

For the QSAM file PAYCHECK, note the record size and block size on the Data Set Characteristics report. The record size is 132 and the block size is 1320. By comparing the record size to the block size (132 records and 1320 blocks), you see that the program writes and reads for each block of ten records.

You may be able to improve performance by increasing the block size of the file to the largest possible for the DASD or tape you are using. In this case study, the block size of the QSAM file PAYCHECK was changed from 1320 bytes to the significantly larger block size of 23364 bytes. With this larger block size, the program will perform fewer reads and writes.

Reviewing the Data Set Characteristics Supplement Report

To further investigate VSAM buffering, refer to the Data Set Characteristics Supplement report, Figure 2-26 on page 2-23. This report shows the OPEN INTENT\PROCESSING MODE, which indicates whether the application opens files for input, output, or both. For VSAM files, it also shows whether the records are processed sequentially or directly (randomly), and provides the number of CIs (control intervals) in a CA (control area).

VSAM Blocking and Buffering

Note on the Data Set Characteristics Supplement report (Figure 2-26 on page 2-23) that for the data portion of the VSAM file EMPLOYEE the access is OUTPUT, DIR, meaning that the file is directly accessed. Recall that on the Data Set Characteristics report (Figure 2-25 on page 2-22), the data portion of the EMPLOYEE file has two buffers but the index portion has one buffer. Although these represent the VSAM defaults, they are not efficient for this particular type of processing.

Figure 2-26. Data Set Characteristics Supplement Report (PAYROLL1)

```
                                ** DATA SET CHARACTERISTICS SUPPLEMENT **

DDNAME   ACCESS     DSNAME                                  OPEN INTENT\PROCESSING MODE
         METHOD

EMPLOYEE VSAM KSDS  WPAV.WPAJEA.PAY01.EMPLOYEE              OUTPUT,DIR
  FILE:  RECFM........                                VSAM:  FREESPACE... 916K    USER RECORDS...        9,999
         EXTENTS......  1                                    SHROPTS....(1,3)    LOGICAL OPERATIONS:
                                                             CI/CA......  150     DELETES........            0
                                                             %CI FREE.....  0     UPDATES........        6,998
                                                             %CA FREE.....  0     RETRIEVES......        6,999
                                                                                  INSERTS........            0

EMPLOYEE VSAM INDEX WPAV.WPAJEA.PAY01.EMPLOYEE              OUTPUT,DIR
  FILE:  RECFM........                                VSAM:  FREESPACE...  24K    USER RECORDS...            4
         EXTENTS......  1                                    SHROPTS....(1,3)    LOGICAL OPERATIONS:
                                                             CI/CA......   10     DELETES........            0
                                                             %CI FREE.....  0     UPDATES........            0
                                                             %CA FREE.....  0     RETRIEVES......            0
                                                             INDEX LVLS.    2     INSERTS........            0

PAYCHECK QSAM       WPAJEA.PAY01.OUTPUT.PAYCHEKS            OUTPUT
  FILE:  RECFM........FBM
         EXTENTS......  4


STEPLIB  BPAM       WPAJEA.USER.LOAD                        INPUT
  FILE:  RECFM........  U
         EXTENTS......  2


TAXDATA  VSAM KSDS  WPAV.WPAJEA.PAY01.TAXDATA               INPUT,DYN
  FILE:  RECFM........                                VSAM:  FREESPACE... 396K    USER RECORDS...           58
         EXTENTS......  1                                    SHROPTS....(1,3)    LOGICAL OPERATIONS:
                                                             CI/CA......  100     DELETES........            0
                                                             %CI FREE.....  0     UPDATES........            0
                                                             %CA FREE.....  0     RETRIEVES......            0
                                                                                  INSERTS........            0

TAXDATA  VSAM INDEX WPAV.WPAJEA.PAY01.TAXDATA               INPUT,DYN
  FILE:  RECFM........                                VSAM:  FREESPACE...  36K    USER RECORDS...            1
         EXTENTS......  1                                    SHROPTS....(1,3)    LOGICAL OPERATIONS:
                                                             CI/CA.......  10     DELETES........            0
                                                             %CI FREE.....  0     UPDATES........            0
                                                             %CA FREE.....  0     RETRIEVES......            0
                                                             INDEX LVLS.    1     INSERTS........            0
```

In this case, performance improved when the number of buffers in the index portion of the VSAM cluster EMPLOYEE was changed from the default of 1 to 2.

Changing the Application

To change the block size in the QSAM file and to change the number of index buffers in the VSAM cluster, the application's original job control language (JCL) statements needed to be changed. The original and the modified JCL statements are shown below. The changed lines are indicated with change bars.

```
$EXEC1:
//JOBX JOB (USERX,WPAOSMAPM),'$EXEC2',CLASS=2,
// MSGCLASS=Q,NOTIFY=WPAKMO,REGION=4096K,TIME=1440
      :
      :
//*************************************************************
//RUNAPP  EXEC PGM=PAYROLL1,COND=EVEN
//STEPLIB   DD DSN=WPAKMO.USER.LOAD,DISP=SHR
//SYSUDUMP  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSOUT    DD SYSOUT=*
//TIMSHEET  DD DSN=WPAV.WPAKMO.PAY01.TIMSHEET,DISP=SHR
//EMPLOYEE  DD DSN=WPAV.WPAKMO.PAY01.EMPLOYEE,DISP=OLD
//TAXDATA   DD DSN=WPAV.WPAKMO.PAY01.TAXDATA,DISP=SHR
//PAYCHECK  DD DSN=&PAYCHEK,DISP=(NEW,CATLG),
//          UNIT=WPAANY,DCB=(RECFM=FBA,LRECL=132,BLKSIZE=1320),
//          SPACE=(CYL,(35,20),RLSE)


$EXEC2:
//*************************************************************
//RUNAPP  EXEC PGM=PAYROLL&LVL,COND=EVEN
//STEPLIB   DD DSN=WPAKMO.USER.LOAD,DISP=SHR
//SYSUDUMP  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSOUT    DD SYSOUT=*
//TIMSHEET  DD DSN=WPAV.WPAKMO.PAY01.TIMSHEET,DISP=SHR
//EMPLOYEE  DD DSN=WPAV.WPAKMO.PAY01.EMPLOYEE,DISP=OLD,
//          AMP=('BUFNI=2')
//TAXDATA   DD DSN=WPAV.WPAKMO.PAY01.TAXDATA,DISP=SHR
//PAYCHECK  DD DSN=&PAYCHEK,DISP=(NEW,CATLG),
//          UNIT=WPAANY,DCB=(RECFM=FBA,LRECL=132,BLKSIZE=23364),
//          SPACE=(CYL,(35,20),RLSE)
```

Measure to Verify the Effects of the Changes

After making the changes to the QSAM and VSAM files, measure again with STROBE. In this measurement, to collect enough data for a valid Profile, the sample size was increased to 20,000. Because STROBE was measuring at a high rate, stretch time (which includes the time that STROBE was collecting data) accounts for most of the run time. Note in the resulting Measurement Session Data report (Figure 2-27 on page 2-25) that wait time dropped to 1.84 seconds in PAYROLL2 from 10 minutes and 2.35 seconds in PAYROLL1. CPU time dropped to 0.32 seconds from 14.18 seconds. Measuring with STROBE showed you where to make simple changes to the application—affecting only a few lines—that substantially improved the application's performance.

Figure 2-27. Measurement Session Data Report (PAYROLL2)

```
                                    ** MEASUREMENT SESSION DATA **

-------- JOB ENVIRONMENT --------        ------- MEASUREMENT PARAMETERS ------        ----- MEASUREMENT STATISTICS ----

 PROGRAM MEASURED   -    PAYROLL2        ESTIMATED SESSION TIME  -       1 MIN        CPS TIME PERCENT -          14.80
 JOB NAME           -    WPAJEAG2        TARGET SAMPLE SIZE      -      20,000        WAIT TIME PERCENT -          85.20
 JOB NUMBER         -    JOB06996        REQUEST NUMBER   (Q)    -         387        RUN MARGIN OF ERROR PCT  -   5.18
 STEP NAME   -             RUNAPP                                                    CPU MARGIN OF ERROR PCT  -  13.46

 DATE OF SESSION    -  07/20/1999        SYS REQ -  TLP01                           TOTAL SAMPLES TAKEN -         358
 TIME OF SESSION    -    16:13:27                                                   TOTAL SAMPLES PROCESSED -     358
 CONDITION CODE     -      C-0000        MODULE MAPPING BASELINE -           2      INITIAL SAMPLING RATE- 333.33/SEC
                                         BASELINE OVERRIDE -            TAXRTN1    FINAL SAMPLING RATE  - 333.33/SEC
 SYSTEM -              ESA SP5.2.0
 DFSMS              -        1.2.0       --------- REPORT PARAMETERS ---------    SESSION TIME -     0 MIN 26.47 SEC
 CPU MODEL          -   3090-600S                                                  CPU TIME -         0 MIN  0.32 SEC
 SMF/SYSTEM ID -        MVST/TLP01        RESOLUTION    -              64 BYTES     WAIT TIME    -     0 MIN  1.84 SEC
                                         SORTSIZE              -       999,999     STRETCH TIME -     0 MIN 24.31 SEC
 REGION SIZE BELOW 16M  -  4,160K        LINES/PAGE            -            60
 REGION SIZE ABOVE  -     32,768K                                                  SRB TIME     -     0 MIN  0.37 SEC
                                         DASD= 2.0%  DASDGAP= 5                     SERVICE UNITS-                336
 PTF LVL- 2.3.0.FS000000/FS000000
                                         DATE FORMAT                MM/DD/YYYY      PAGES IN-      0     OUT-       0
                                         TIME FORMAT (24 HOURS)       HH:MM:SS      PAGING RATE  -           0.00/SEC
                                                                                    EXCPS -         21       0.79/SEC


 SAMPLE DATA SET  - STROBE.V1.WPAJEAG2.S002D001
```

Additional Opportunities for Improving Performance

One Profile will rarely show every possible type of performance improvement opportunity. The following section discusses examples that show additional types of performance improvement opportunities. In the cases below, assume that you have discovered high wait time in a valid Measurement Session Data report (not shown). The next report to examine is the Resource Demand Distribution report.

Looking for Contention

Figure 2-28 on page 2-25 shows the Resource Demand Distribution report for a program that reads two files.

Figure 2-28. Resource Demand Distribution Report (Internal Contention)

```
                                                ** RESOURCE DEMAND DISTRIBUTION **

                          ---- PERCENT OF RUN TIME -----     ----- PERCENT OF RUN TIME SPENT -------   CUMULATIVE PERCENTAGES
TASK OR    RESOURCE        SERVICED   SERVICED  SERVICED      SOLO      SOLO      SOLO      CAUSING        SOLO      CAUSING
 DDNAME                     BY CPU     BY I/O  BY EITHER     IN CPU    IN I/O  IN EITHER   CPU WAIT        TIME    CPU WAIT

TESTCONT   CPU               7.52        .00       7.52       7.15       .00       7.15        .15         7.15         .15
INFILE1    3380               .23      44.24      44.47        .15     44.09      44.24      51.77        51.39       51.92
INFILE2    3380               .23      35.21      35.21        .00     34.61      34.84      37.85        86.23       89.77
.FILEMGT                      .30       5.79       6.09        .23      5.79       6.02       1.88        92.25       91.65
STEPLIB    3380               .00        .08        .08        .00       .08        .08        .08        92.33       91.73
```

When considered together, processing related to files INFILE1 and INFILE2 consumes most of the run time, as shown in the column SERVICED BY I/O. The INFILE1 value is 44.24 and the INFILE2 value is 35.21. Added together, these values equal 79.45%. In the CAUSING CPU WAIT column, the value for INFILE1 is 51.77 and the value for INFILE2 is 37.85, which when added together equals 89.62%. Some of this wait may result from other applications' access to the devices that contain the data. To determine whether the additional wait results from two or more files being accessed at the same time, refer to the Time Distribution of Activity Level report, Figure 2-29 on page 2-26.

Figure 2-29. Time Distribution of Activity Level Report (Internal Contention)

```
                                  ** TIME DISTRIBUTION OF ACTIVITY LEVEL **

TASK OR   RESOURCE   N X 10 PLUS OR MINUS 5 IS PERCENT OF FULL UTILIZATION   * IS GREATER THAN 95%     - IS LESS THAN 5%
 DDNAME           .-------------------------------------------------------------------------------------------------------.
TESTCONT  CPU     .22211211   21122 2  1121111111 21 12   111  1 12 11 1 11 11211 112113 121 111 12121 111111221  11 21.
INFILE1   3380    .2743434555435525635555456553546452356345258783875744926848443525445444665555545553245635544453555555  .
INFILE2   3380    .4226235344544563254455334355534464426335233 6223254 52161443455433455424532245434553543553535442422 .
.FILEMGT          .  1 22 1 121    21   1  1 111 1 12 1 2221  2   121 1 1   3 112 11 1     1111   11  1  21     11  257.
STEPLIB   3380    .1                                                                                                    .
                  0----0----1----1----2----2----3----3----4----4----5----5----6----6----7----7----8----8----9----9----*.
                  0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----*.
                  START RUN                PERCENT OF ALLOCATED RUN TIME                                    END RUN
```

The Time Distribution of Activity Level report shows activity among the resources used by the application in 100 vertical columns where each column represents a 1% time slice. In each of these vertical slices, there is almost equal access to INFILE1 and INFILE2, indicating that the application file I/O activities overlap. That is, they occur in the same time slice but not necessarily simultaneously.

If multiple files showing high wait are being accessed at the same time, turn to the I/O Facility Utilization Summary report, Figure 2-30 on page 2-26, to see if the files are located on the same volume. If they are, the wait time may be caused by internal device contention. Notice that both files reside on volume WPA003. To eliminate the internal contention, move one of the files to a separate volume.

Figure 2-30. I/O Facility Utilization Summary Report (Internal Contention)

```
                                  ** I/O FACILITY UTILIZATION SUMMARY **

 UNIT       DEVICE CACHE       VOLUME     DDNAME   I/O        RUN TIME PERCENT     RUN TIME HISTOGRAM  MARGIN OF ERROR
2.69%
   NO         TYPE   ELIG        ID                              SOLO     TOTAL     .00     11.50     23.00     34.50
     46.00

 2E8     DA 3380             WPA003   .FILEMGT            4.82       4.82  .****
 2E8     DA 3380             WPA003   INFILE1    I       44.09      44.24  .****************************************
 2E8     DA 3380             WPA003   INFILE2    I       34.61      35.21  .*******************************
                                                         -----      -----
UNIT  2E8 TOTALS                                         83.52      84.27
```

Indexed Performance Profiles

Indexing is the optional process of associating the measurement data that STROBE collects in the sample data set with specific line numbers in the application program's source code and procedure names. If you do not index your Performance Profile, STROBE will format the reports without source code information as shown on Figure 2-31 on page 2-27.

Being able to refer to specific line numbers and procedure names in the previous section made it easier to interpret the Performance Profile and determine the action necessary to improve performance. To index the Performance Profile, create a map data set (which essentially maps the layout of the compiled source code) and include it when you produce the Performance Profile. You can also use DDIO files created by the Compuware language processor. The resulting STROBE Performance Profile is indexed—the reports in the Performance Profile relate the location of concentrations of CPU activity or wait time to your application's code statements, procedure names, and line numbers. Because it makes Performance Profiles easier to interpret, Compuware strongly recommends that you create and use map data sets or DDIO files.

As defined earlier, a control section is the smallest unit of execution produced by a compiler or assembler that can be linked or replaced in a load module. When you compile an application program, the compiler creates one or more control sections corresponding to the statements in the source code. The next section describes in greater

detail what a STROBE report for unindexed control sections looks like. It then discusses the format of reports for indexed control sections and describes the language-specific rules that govern indexing.

For more explicit instructions on the language-specific options that you need to specify, see the sections that discuss indexing in Chapter 2, "Interpreting the STROBE Performance Profile" and Chapter 4 of the STROBE *STROBE MVS User's Guide* or Chapter 2 and Chapter 5 of the STROBE *STROBE MVS User's Guide with Advanced Session Management.*

Note: To ensure that the control section information is complete, you must compile all VisualAge products with the binder option EDIT=YES (the default).

Unindexed Control Sections

When you do not index source code measured in an application, each line entry details activity within a *codeblock,* which is an area with a size (in bytes) that is equal to the report resolution. You can specify the report resolution when you produce the Performance Profile. In an unindexed Performance Profile, line number and procedure name fields are blank. The unindexed Program Usage By Procedure report, Figure 2-31 on page 2-27, has the default report resolution (shown in the INTERVAL LENGTH Column) of 64 bytes.

Figure 2-31. Unindexed Control Section Subreport

```
                                    ** PROGRAM USAGE BY PROCEDURE **

MODULE - STRBSAM1                          SECTION - SAMPLEW

 LINE    PROCEDURE          STARTING      INTERVAL  CPU TIME PERCENT      CPU TIME HISTOGRAM  MARGIN OF ERROR: 6.30%
NUMBER     NAME             LOCATION        LENGTH   SOLO     TOTAL       .00      7.00     14.00     21.00    28.00

                            000000            1984    .00       .00      - .
                            0007C0              64    .41       .41        .
                            000800              64    .41       .41        .
                            000840              64    .41       .41        .
                            000880              64    .83       .83        .*
                            0008C0              64    .00       .00        .
                            000900              64    .41       .41        .
                            000940              64   6.20      6.20        .********
                            000980              64  24.79     24.79        .***********************************
                            0009C0              64  26.45     26.45        .************************************
                            000A00              64  12.40     12.40        .*****************
                            000A40              64  16.94     16.94        .************************
                            000A80              64   3.72      3.72        .*****
                            000AC0             576    .00       .00      - .
                                                      -----     -----
SECTION  SAMPLEW   TOTALS                             92.97     92.97
```

The Program Usage by Procedure report combines entries for inactive codeblocks that are adjacent. In this case, the resulting entry covers an area whose size in bytes is greater than the report resolution. This is noted with a dash to the left of the histogram column as shown in Figure 2-31 on page 2-27. The example has entries with codeblocks of sizes of 1984 and 576 bytes.

You can still use the information in an unindexed report to determine what statements are causing the activity. Follow these steps:

- Produce a listing of the compilation of the source code that matches exactly the compile options for the program executed. Compile it with the option to produce a procedure map that relates the statement number and text with the offset.
- In the procedure map section of the listing, look up the statement number and procedure name that contains the offset.
- Next, look in the source statement section for the exact statement code.

Indexed Control Sections

Indexing is a way to speed up your analysis. To index a Performance Profile, you produce a map data set or use a DDIO file. Each procedure comprises one or more consecutive source program statements; the procedure or group of statements is defined according to the rules for the source language.

If a compiler has produced code not associated with any source program statements, indexing assigns a procedure name that suggests the function of the code but leaves the line number field blank. In Figure 2-32 on page 2-28, INIT1 CODE is an assigned procedure name.

CPU activity always appears for a range of source lines beginning with the number listed in the line column. For example, if the statements span an area that is less than the report resolution, the activity will be rolled up into the first statement. In Figure 2-32 on page 2-28, activity for line 71 is combined into line 70. STROBE suppresses partitioning when consecutive areas are inactive. CPU usage is reported for a group of lines beginning at the line number reported, not for the specific line number.

The language-specific rules that govern indexing, described in the following sections, specify how the STROBE language-specific indexing program (called an *Indexer)* identifies procedure boundaries, assigns procedure names, and partitions long procedures into codeblocks for reporting.

COBOL

The STROBE COBOL Feature provides indexing for COBOL code. The COBOL language Indexer treats each paragraph as a procedure, as shown in Figure 2-32 on page 2-28. It also treats statements that follow a section name but precede a paragraph as procedures. It assigns the paragraph name and the section name to the procedure names. When one or more section names appear before a paragraph name with no code-generating statements in between, a break in the report identifies the section name with the lowest line number.

Figure 2-32. Indexed Control Section Subreport—COBOL

```
MODULE - STRBSAM1                              SECTION - SAMPLEA
                                               SOURCE LANGUAGE - ANS COBOL VS                                VC24X25

 LINE     PROCEDURE      STARTING    INTERVAL    CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  6.30%
NUMBER       NAME        LOCATION      LENGTH     SOLO     TOTAL        .00      6.50     13.00     19.50     26.00

        INIT1 CODE        000000         1998      .00       .00      - .
    54  MAIN-LINE         0007CE           96      .83       .83        .*
    57     PERFORM        00082E           48      .41       .41        .
    58     PERFORM        00085E           48      .00       .00        .
    59  MAIN-3            00088E           90      .83       .83        .*
    65     WRITE          0008E8           56      .41       .41        .
    65     WRITE          000920           74      .00       .00      - .
    70  WAY1-SUBSCRIPT    00096A           54    14.88     14.88        .**********************
    72     ADD            0009A0           40    17.77     17.77        .**************************
    76  WAY2-INDEXED      0009C8           48    24.38     24.38        .************************************
    79     IF             0009F8           28     5.37      5.37        .********
    82  WAY3-ABS-SUB      000A14           54    10.74     10.74        .****************
    86     ADD            000A4A           42    11.57     11.57        .*****************
    89  WAY4-STRAIGHT     000A74           36     4.55      4.55        .*******
    95  END-JOB           000A98          338     1.24      1.24        .*
    97     STOP           000BEA          270      .00       .00      - .
                                                -----     -----
SECTION  SAMPLEW  TOTALS                        92.98     92.98
```

When the Indexer partitions a COBOL procedure into codeblocks for reporting, it does so at a COBOL verb and uses the verb as a label for the codeblock.

Assembler

The assembler Indexer treats a sequence of source text statements that begin with a labeled executable statement and end with the statement preceding the next labeled executable statement as a procedure. It uses the symbol appearing in the name field of the first statement as the procedure name. For example, in Figure 2-33 on page 2-29, the assembler indexer treats BASE and SAVEREGS as procedures. If two or more symbols are assigned to the same location, the Indexer assigns to the procedure name the symbol appearing in the name field of the statement with the highest line number. The Indexer also considers an unlabeled sequence of statements immediately following an ORG statement as a procedure. For such a sequence, it creates a procedure name containing three asterisks and the operation code in the first statement that follows the ORG statement.

Figure 2-33. Indexed Control Section Subreport—Assembler

```
MODULE - SAMPLE                                SECTION- SAMPLEW
                                               SOURCE LANGUAGE - ASSEMBLER VS                                    VC01X04

 LINE     PROCEDURE         STARTING   INTERVAL      CPU TIME PERCENT      CPU TIME HISTOGRAM  MARGIN OF ERROR  - 1.98%
NUMBER      NAME           LOCATION     LENGTH       SOLO     TOTAL       .00      7.50     15.00     22.50      30.00

    17  BASE                000000          18         .00       .00       .
    22  SAVEREGS            000012          50         .00       .00       .
    40  INITIAL             000044          52         .00       .00       .
    55     LA               000078          20         .44      1.76       .*++
    62  CALC01              00008C          58         .00       .75       .*
    81  CALC02              0000C6          26       17.21     28.02       .************************+++++++++++++++
    89     MVC              0000E0          58         .31      1.53       .**
   108  ALTER01             00011A          26         .05       .17       .
   116  ALTER02             000134          80         .02       .24       .
   144  CHECK01             000184          18        6.10     16.81       .********++++++++++++++++
   151  CHECK02             000196           4        3.87      6.09       .******+++
   152  CHECK03             00019A         194        8.85     11.10       .************+++
   163  ABEND01             00025C          16        1.72      1.97       .**+
   168  FINIS               00026C          20        4.13      4.35       .*****
   174  RETRY01             000280          30        3.24      3.24       .****
   186  RETRY02             00029E          18         .47       .69       .*
   191  SAVEAREA            0002B0          12         .00       .00       .
   195  WORKAREA            0002BC         472         .00       .00       -.
                                                     -----     -----
  SECTION   SAMPLEW  TOTALS                         46.41     76.72
```

When the Indexer partitions an assembler language procedure into codeblocks for reporting, it creates a partition immediately following a branch instruction. It identifies the codeblock by the operation code of the first instruction that follows the branch. For example, in Figure 2-33 on page 2-29, the indexer partitions the report at the instructions LA (load address) and MVC (move character).

FORTRAN

The STROBE FORTRAN Feature provides indexing for FORTRAN code. The FORTRAN language Indexer treats as a procedure a sequence of source text statements beginning with a numbered executable statement and ending with the statement preceding the next numbered executable statement. The FORTRAN Indexer uses the statement number, preceded by STMT. LABEL, as the procedure name (Figure 2-34 on page 2-30).

Figure 2-34. Indexed Control Section Subreport—FORTRAN

```
MODULE - SAMPLE                                SECTION - SETCODE
                                               SOURCE LANGUAGE - VS FORTRAN                                    VF111X6

  LINE    PROCEDURE            STARTING   INTERVAL      CPU TIME  PERCENT    CPU TIME HISTOGRAM  MARGIN OF ERROR - 1.98%
NUMBER      NAME               LOCATION    LENGTH        SOLO     TOTAL     .00    3.50      7.00     10.50    14.00

        ENTRY CODE  MAIN        000000         36         .02       .02     .
        DATA AREA               000024        388         .00       .00     .
        CODE AREA               0001A8         48         .02       .02     .
  0029  STMT. LABEL 50          0001D8         18         .02       .02     .
  0032  STMT. LABEL 3           0001EA          8         .00       .00     .
           INT. 07.001          0001F2         30       13.22     13.22     .************************************
  0034  STMT. LABEL 2           000210         88        9.78      9.78     .**************************
  0037  STMT. LABEL 4           000268         12         .00       .00     .
  0038  STMT. LABEL 5           000274          8         .00       .00     .
  0039  STMT. LABEL 10          00027C          8         .00       .00     .
        EPILOGUE                000284         36         .04       .04     .
        PROLOGUE                0002A8        114         .02       .02     .
                                                         -----     -----
   SECTION   SETCODE  TOTALS                              23.14     23.14
```

When the FORTRAN Indexer partitions a FORTRAN procedure into codeblocks for reporting, it labels the codeblocks with compiler-generated internal statement labels, such as “INT. 07.001”. These statement numbers generally correspond to targets of internally generated branch operations.

PL/I

The STROBE PL/I Feature provides indexing for PL/I code. The PL/I language Indexer treats as a procedure a sequence of statements beginning with a labeled statement or with an executable statement to which control can be passed, and continuing to the next such statement (Figure 2-35 on page 2-30). In general, procedure boundaries other than statement labels coincide with the beginning and ending boundaries of BEGIN and ON-UNIT blocks, PL/I procedures, DO groups, and IF statement alternatives. The PL/I Indexer uses the label associated with a boundary for the procedure name and gives other boundaries one of the following labels, depending on the function of the code before or after the boundary:

```
STATIC PROLOGUE
DO- OR IF- BOUNDARY
ON UNIT BLOCK
ON UNIT BLOCK END
BEGIN BLOCK
BLOCK/PROC BOUNDARY
PL/I EPILOG
```

Figure 2-35. Indexed Control Section Subreport—PL/I

```
MODULE - SAMPLE                                SECTION - **INTRST
                                               SOURCE LANGUAGE - IBM PL/I OPT                                  VP30X14

  LINE     PROCEDURE          STARTING   INTERVAL     CPU TIME PERCENT      CPU TIME HISTOGRAM  MARGIN OF ERROR :   1.98%
NUMBER       NAME             LOCATION    LENGTH       SOLO     TOTAL       .00     10.00      20.00    30.00      40.00

    50  INTEREST               000000        438        .00       .00     - .
   170  TEST_COMPUTE           0001B6         20        .63       .63       .*
   220     STATEMENT-NUMBER    0001CA         24      38.67     38.67       .****************************************
   230     STATEMENT-NUMBER    0001E2        184      31.02     31.02       .*******************************
   250  DO- OR IF- BOUNDARY    0001EC         12        .21       .21       .
   280  DO- OR IF- BOUNDARY    0002A6         80        .00       .00       .
   290     STATEMENT-NUMBER    0002F6         80        .21       .21       .
   310     STATEMENT-NUMBER    000346         20        .00       .00       .
   320  DO- OR IF- BOUNDARY    00035A        160        .00       .00       .
   360  BLOCK/PROC BOUNDARY    0003FA          2        .00       .00       .
                                                      -----     -----
  SECTION   **INTRST TOTALS                           70.73     70.73
```

When the Indexer partitions a PL/I procedure for reporting, it partitions at the beginning of a PL/I statement and uses the expression STATEMENT-NUMBER as its label.

C Language

The STROBE C Feature provides indexing for C code. The C language Indexer identifies code as either SAS/C or IBM C/370. All C language programs begin with a function named *main*. The Indexer identifies C source statement numbers associated with a particular codeblock. For example, in Figure 2-36 on page 2-31, activity from hexadecimal locations 0000C4 at line 32 to 000107 at line 40 is reported under next-nod=node.

Compilers for IBM C/370 and SAS/C treat unnamed control sections differently. The SAS/C compiler produces function names associated with control sections, as shown in Figure 2-36 on page 2-31.

Figure 2-36. Indexed Control Section Subreport—SAS/C

```
MODULE - CPDSSAS                                SECTION- MAIN@
                                                SOURCE LANGUAGE - SAS C                                                    VC45X00

 LINE      PROCEDURE           STARTING      INTERVAL  CPU TIME PERCENT        CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.81%
NUMBER        NAME             LOCATION       LENGTH   SOLO     TOTAL         .00       .50      1.00      1.50      2.00

000005  main                    000000            196    .00       .00        -.
000032     next_nod=node;       0000C4             68    .99       .99         .******************
000041     free_mem(node);      000108            346    .00       .00        -.
000114     return(NULL);        000262             86    .00       .08         .+
000119     perror("fread:");    0002B8            212    .00       .00        -.
000156  gen_node                00038C             62    .00       .08         .+
000175     char *ptr, *name;    0003CA             74    .08       .25         .*+++
000185     list_end = 1;        000414             32    .00       .08         .+
000198     info_byte = (unsi    000434             66    .00       .08         .+
000202     ptr += skip;         000476             46    .00       .08         .+
000212  *add_name               0004A4             66    .00       .25         .+++++
000220     newnode = malloc(    0004E6             70    .00       .16         .+++
000223     exit(-1);            00052C             30    .00       .00         .
000230     newnode-next = N     00054A             54    .08       .49         .*++++++++
000250  free_mem                000580            108    .00       .00        -.
                                                        -----     -----
SECTION  MAIN@     TOTALS                                1.15      2.54
```

When generating object code, however, the IBM C/370 and AD/Cycle C/370 compilers can generate unnamed control sections for the program. Figure 2-37 on page 2-31 shows an example of an indexed report for IBM C/370. A compiler directive #PRAGMA CSECT forces the IBM compiler to generate a control section with a name that you supply. If you do not specify that directive, the compiler generates unnamed control sections, which STROBE identifies as $PRIVATE when it measures.

Figure 2-37. Indexed Control Section Subreport—IBM C/370

```
MODULE - CPDSIBM                                SECTION - cpdsibm
                                                SOURCE LANGUAGE - IBM C/370                                                VC12X00

 LINE      PROCEDURE           STARTING      INTERVAL  CPU TIME PERCENT        CPU TIME HISTOGRAM  MARGIN OF ERROR:  3.71%
NUMBER        NAME             LOCATION       LENGTH   SOLO     TOTAL         .00       .50      1.00      1.50      2.00

000000  Compile Unit Block      000000            372    .00       .00       - .
000579     next_node = list.    000174             36    .29       .29        .*****
000583     free_mem(node);      000198            638    .00       .00       - .
000657     if ((bytes != siz    000416            122    .14       .14        .**
000662     exit(-1);            000490            562    .00       .00       - .
000737     info_byte = (unsi    0006C2             70    .43       .57        .*********+++
000740     skip = (info_byte    000708             56    .00       .14        .++
000000  Start of Epilog         000740            172    .00       .00       - .
000000  Prolog add_name         0007EC             86    .14       .14        .**
000751  *add_name               000842             70    .29       .29        .*****
000762     exit(-1);            000888             36    .00       .00       - .
000768     newnode-nameNAM      0008AC             54    .14       .14        .**
000776     *last_ptr = newno    0008E2             54    .29       .29        .*****
000000  Start of Epilog         000918            368    .00       .00       - .
                                                        -----     -----
SECTION  cpdsibm   TOTALS                             1.72      2.00
```

When the C Indexer detects multiple unnamed control sections, it differentiates them by naming the second control section $PRI0000 and adding one to the four-digit suffix for

each subsequent control section. When the C Indexer maps the compiled program, it uses the generated names rather than the source function names.

Additional Indexers

The STROBE ADABAS/NATURAL, CA-IDMS, CSP, and COOL:Gen Features provide additional indexers that help relate resource usage to lines of code. The STROBE Java Feature provides index-level information without employing an indexer. For more information, see the appropriate STROBE Feature manual.

Chapter 3.
The STROBE Performance Profile

The STROBE Performance Profile is a collection of reports that provides comprehensive information about the performance of a program or subsystem during a measurement session. The Performance Profile enables you to find those sections of a program or subsystem that offer significant opportunities for performance improvement. This chapter defines key report terminology and provides a reference for each report in the Performance Profile. It lists the reports in the order they appear in the Performance Profile and describes each field and column that the report displays.

Report Terminology

The STROBE Performance Profile reports use the terms supplied in the following sections to measure resource use.

Terms Displayed on the Performance Profile Reports

Session time — The amount of time (wall clock time) of the measurement session. It includes CPU time, wait time, and stretch time.

CPU time — The amount of time during the measurement session that one or more CPUs were executing tasks in the measured job step, exclusive of the measurement task itself, expressed in minutes and seconds to the nearest hundredth. Since you are usually billed for the CPU time used, it is considered the most important measure of system resource time.

CPS time percent — The percentage of time during which the central processing subsystem (comprising one or more CPUs) was in use by application tasks executing within the measured job step. STROBE computes the value as the ratio of samples in which it observed one or more sites of execution to the total number of samples taken.

Stretch time — The estimated amount of time that the CPU was unavailable to process programs executing in the measured address space because of demands made by higher-priority address spaces and by service request block (SRB) processing time for all address spaces. Stretch time may be high when multiple logical partitions (LPARs) are executing on the same complex of processors.

The value in the STRETCH TIME field reported on Measurement Session Data reports can vary widely, even when you run STROBE under what seem to be identical conditions. This may result from a variation in system load.

SRB time — The amount of time during which the system was executing service request blocks (SRBs), which represent high-priority service routines.

Wait time The portion of run time during which no task within the measured address space was able to make use of the CPU time available to it. Wait time usually occurs when a step is waiting for I/O to complete, an operator to respond, a user to provide input, or data to come from an allied address space.

Total time and Solo time STROBE shows percentages for both total and solo resource usage time.

Total time is a percentage of time that represents all the time during which the activity used a resource. This percentage includes all CPU execution time and all wait time.

Solo time is the percentage of total time during which STROBE detected one and only one activity for the measured job step. STROBE measures solo time to highlight instances of resource use that have a direct effect on the run time of a batch processing program.

In the case of task execution activity, solo time is the percentage of time during which only one CPU serviced the measured job step, with no concurrent file access activities. For file access activities, STROBE reports both CPU and I/O time. CPU time is the percentage of time spent in operating system data management modules. The I/O time reported is the percentage of time a device serviced a file access activity. Therefore, solo time can be either CPU time or I/O facility time spent servicing a file access activity, so long as neither task execution activity nor any other file access activity was serviced within the address space during that time.

Run time Is the amount of time during which STROBE measured. It includes CPU time, comprising one or more CPUs, and wait time.

Pseudo-Entities

During its execution, the program or subsystem spends some portion of CPU time in supervisory routines that service components of the operating system control program or subsystem. Instead of reporting in detail on each such routine, STROBE groups them by function and reports the CPU time spent performing each function as time spent within a STROBE pseudo-entity.

The name of a pseudo-entity always begins with the character "." and reflects the function performed collectively by the routines it represents. STROBE defines four classes of pseudo-entities:

- *pseudo-activity*—The pseudo-activity .FILEMGT groups file management operations such as opening and closing files.
- *pseudo-module*—The pseudo-module .SYSTEM groups all supervisory functions.
- *pseudo-transaction*—A pseudo-transaction combines online subsystem overhead functions that STROBE cannot assign to any specific transaction. For example, the pseudo-transaction .CICS includes all CICS supervisory functions that STROBE could not assign to a user-defined transaction.
- *pseudo-section*—A pseudo-section is a collection of either
  - dynamically linked load modules, such as .IOCS (data management modules), or
  - compiler library subroutines that are control sections within the target program load module, such as .COBLIB (COBOL library routines).

  Pseudo-sections are subsets of either a real load module or the pseudo-module .SYSTEM. Figure 3-1 on page 3-3 shows a Program Usage By Procedure report that

displays pseudo-sections. The report lists the module's name or the SVC's number, the control section name (if STROBE obtained one during sampling), and a brief description of its function. Normally, a report does not show detailed module or control section information for system routines. You can, however, obtain such a breakdown for selected modules by using the DETAIL parameter on the STROBE - DETAIL FOR A PERFORMANCE PROFILE STROBE/ISPF panel and supplying the module names to STROBE during reporting. See the *STROBE MVS User's Guide* Chapter 2 and Chapter 4 or the *STROBE MVS User's Guide with Advanced Session Management* Chapter 2 and Chapter 5 for more detailed information.

See Appendix A, "Program Structure and STROBE Pseudo-Entities" for a description of all STROBE pseudo-entities.

Figure 3-1. Control Section Subreport—Pseudo-Sections

```
                                        ** PROGRAM USAGE BY PROCEDURE **

.SYSTEM   SYSTEM SERVICES                      .IOCS     DATA MANAGEMENT SERVICES

MODULE   SECTION        FUNCTION           INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  6.30%
  NAME      NAME                            LENGTH   SOLO     TOTAL        .00       .50      1.00      1.50      2.00

IEC0SCR1           PHYSICAL IOCS              3418   1.24      1.24        .************************
IGG019AI           QSAM SMPL PUT LOCATE F/U    144    .83       .83        .****************
IGG0193B           QSAM PUT                  25608    .41       .41        .********
                                                     -----     -----
         .IOCS     TOTALS                           2.48      2.48

        .SYSTEM   SYSTEM SERVICES                      .SVC      SUPERVISOR CONTROL
MODULE   SECTION        FUNCTION           INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  6.30%
  NAME      NAME                            LENGTH   SOLO     TOTAL        .00       .50      1.00      1.50      2.00
SVC 013            TERMINATION                      1.24      1.24        .************************
SVC 019            OPEN                             1.65      1.65        .*********************************
SVC 020            CLOSE                             .41       .41        .********
SVC 083            SMF                               .41       .41        .********
SVC 130            RACHECK                           .41       .41        .********
                                                    -----     -----
         .SVC      TOTALS                           4.12      4.12
```

STROBE Performance Profile Reports

This section discusses all reports in the Performance Profile and provides a comprehensive description of their columns and fields.

Performance Profile Identification

Each page of the Performance Profile bears a running title that, in addition to identifying it as a part of the Performance Profile, contains:

- a subtitle, either user-specified or, by default, the name of the load module executed in the job control statements for the measured step
- the date the Performance Profile was created
- the page number

Measurement Session Data Report

The Measurement Session Data Report enables you to validate the Performance Profile and helps you focus your effort to improve application performance. The Measurement Session Data report (Figure 3-2 on page 3-4) contains four sections:

- Job Environment
- Measurement Parameters

- Measurement Statistics
- Report Parameters

Figure 3-2. Measurement Session Data Report

```
                                    ** MEASUREMENT SESSION DATA **

------- JOB ENVIRONMENT --------      ------- MEASUREMENT PARAMETERS ------      ----- MEASUREMENT STATISTICS ----

PROGRAM MEASURED   -    IKJEFT01      ESTIMATED SESSION TIME  -       5 MIN      CPS TIME PERCENT -           0.08
JOB NAME           -      WPAMXR      TARGET SAMPLE SIZE      -      25,000      WAIT TIME PERCENT -         99.92
JOB NUMBER         -    TSU02748      REQUEST NUMBER   (A)    -         253      RUN MARGIN OF ERROR PCT  -    .62
STEP NAME   -  $WPADB31.$WPAPROC      FINAL SESSION ACTION    -        QUIT      CPU MARGIN OF ERROR PCT  -  22.48

DATE OF SESSION    -  01/11/1999      REQUEST GROUP           -    TEST_GRP      TOTAL SAMPLES TAKEN -      25,000
TIME OF SESSION    -    11:32:09      CONCURRENT SET NAME     -      TEST01      TOTAL SAMPLES PROCESSED -  25,000
                                      CONCURRENT SET ELEMENTS - (T)WPATEST1      INITIAL SAMPLING RATE-  83.33/SEC
                                         WPADATI      WPATRANT                 FINAL SAMPLING RATE  -  83.33/SEC

SYSTEM -             ESA SP5.2.0      SYS REQ - TLP01                          SESSION TIME -   11 MIN  9.05 SEC
DFSMS              -       1.2.0                                               CPU TIME -        0 MIN  0.21 SEC
CPU MODEL          -   3090-600S                                               WAIT TIME    -    4 MIN 36.10 SEC
SMF/SYSTEM ID      -  MVST/TLP01      --------- REPORT PARAMETERS ---------    STRETCH TIME -    6 MIN 32.74 SEC


REGION SIZE BELOW 16M  -  6,208K      REPORT RESOLUTION    -       64 BYTES      SRB TIME     -    0 MIN  4.82 SEC
REGION SIZE ABOVE  -     32,768K      SORTSIZE             -      999,999        SERVICE UNITS-                221
                                      LINES/PAGE           -           60
PTF LVL- 2.3.0.FS000000/FS000000                                                PAGES IN-      0     OUT-       0
                                      DASD= 2.0%  DASDGAP= 5                    PAGING RATE  -           0.00/SEC
                                                                                EXCPS -         12       0.02/SEC
                                      DATE FORMAT               MM/DD/YYYY
                                      TIME FORMAT (24 HOURS)      HH:MM:SS
```

AutoSTROBE Message

If the measurement was initiated by the AutoSTROBE function of the Advanced Session management Feature, a message appears on the top of the Measurement Session Data report indicating the reason AutoSTROBE initiated the measurement. The reason could be: excessive elapsed time, excessive TCB time, or abnormal I/O activity. For more information about AutoSTROBE, see the AutoSTROBE chapter in the *STROBE MVS User's Guide with Advanced Session Management*.

Job Environment

The Job Environment section enables you to confirm that the Performance Profile represents the job and program you intended to measure. It provides additional information, such as system identification and model number of the CPU, which can help you diagnose errors.

PROGRAM MEASURED

The name of the program in which execution of the target program or subsystem was initiated.

JOB NAME

The name of the job under which the target program or subsystem was executed.

JOB NUMBER

The Job Entry Subsystem (JES) number of the target job.

STEP NAME

The name of the job step, or of the procedure step and job step, in which the program or subsystem was executed. For nested procedures, the procedure step name in the invoking procedure and the step name of the invoked procedure appear. For further information about specifying the job step, see the *STROBE MVS User's Guide* Chapter 2 and Chapter 3 or the *STROBE MVS User's Guide with Advanced Session Management* Chapter 2 and Chapter 7.

DATE OF SESSION

The calendar date on which STROBE began measurement. The format of the date depends on the date format specified by the STROBE systems programmer in the STROBE parameter data set.

TIME OF SESSION

The time of day at which STROBE began measurement. The format of the time depends on the time format specified by the STROBE systems programmer in the STROBE parameter data set.

CONDITION CODE/COMPLETION CODE

The condition code or the completion code. If the target program terminated normally while it was being measured, this field shows the condition code. If the target program terminated abnormally, this field shows the completion code. The field does not appear if the measurement completes before the target program completes.

SYSTEM

The version and release level of the operating system that executed the target program or subsystem. Sometimes, to understand all aspects of performance you need to know the specific operating system version and release level.

DFP (or) DFSMS

The version, release, and modification level of DFP (Data Facility Product) or DFSMS (Data Facility Storage Management Subsystem) under which the measurement was performed.

SUBSYSTEM

The name of the subsystem and the parameters chosen when STROBE uses a feature to measure a subsystem. Subsystem-specific reports appear only if this field contains the subsystem information. If you measured a subsystem application, verify that STROBE collected the appropriate subsystem data.

CPU MODEL

The model number of the central processing unit (CPU) that executed the target program or subsystem. Sometimes, to understand certain aspects of performance you need to identify the manufacturer's CPU timing information for the specific CPU identified in the measurement session.

SMF/SYSTEM ID (or) SYSTEM ID

The System Management Facility (SMF) ID and MVS system name on which the measurement session was performed. If you are not using STROBE in a multisystem environment, the label SYSTEM ID appears with the value of the four-character SMF ID.

COUPLING FACILITY NAMES

For systems connected to MVS coupling facilities, the coupling facility names that were in use system-wide during the measurement. This field records a maximum of 32 coupling facility names.

LPAR

For OS/390 systems running in LPAR mode, the name of the logical partition in which the target job executed.

64-BIT ARCHITECTURE ENABLED

Indicates that the zSeries operating system 64-bit architecture has been enabled at IPL time.

REGION SIZE BELOW 16M

The size of the region below the 16-megabyte address line available during the measurement session.

REGION SIZE ABOVE

The size of the region above the 16-megabyte address line available during the measurement session.

PTF LEVEL

The STROBE version and release number, and the number of the highest Program Temporary Fix (PTF) applied to the STROBE system.

The first number shows the highest PTF number for modules executed during the measurement session, during reporting, or during indexing. The second number (following the slash) shows the PTF level that was supplied on your distribution tape. These numbers are updated when a PTF update tape is applied.

SAMPLE DATA SET

The name STROBE created and recorded internally for the sample data set. If you renamed the sample data set, the new name does not appear.

Measurement Parameters

This section identifies the parameters that STROBE applied to the request, specified by the user with an ADD, a CHANGE, or a SEND command, or on the appropriate STROBE/ISPF panel. Verify this section to ensure that STROBE applied the correct values to the measurement and attempted to collect the appropriate data. Some of these fields are optional—they appear only if you specify the associated operands on the appropriate STROBE/ISPF panel or on the ADD operation.

ESTIMATED SESSION TIME

The estimated duration of the measurement session, in minutes.

TARGET SAMPLE SIZE

The selected sample size. This value is the number of samples you request STROBE to take during the estimated session time.

REQUEST NUMBER

The number that STROBE assigned to the measurement request. (A) indicates an active request; (Q) indicates a queued request. An active request is a measurement request for a program that, at the time of the request, was currently executing. A queued request is a measurement request for a program that, at the time of the request, was not yet executing.

FINAL SESSION ACTION

The action that STROBE took when it reached the target sample size in its last measurement session. It gives the number of sample data sets created (blank if 1) followed by the final session action. The following session actions appear if you requested the measurement through STROBE/ISPF or specified the action through the STROBE command language:

- QUIT (stop sampling and terminate the request)
- STOP (stop sampling and suspend the request)

REQUESTED START DATE

For scheduled measurement requests, the date you specified for the measurement to be submitted. The format of the date depends on the date format specified by the STROBE systems programmer in the STROBE parameter data set. This field appears only if you are using the scheduling function of the STROBE Advanced Session Management Feature.

REQUESTED START TIME

For scheduled measurement requests, the time you specified for the measurement request to be submitted. The format of the time depends on the time format specified by the STROBE systems programmer in the STROBE parameter data set. This field appears only if you are using the scheduling function of the STROBE Advanced Session Management Feature.

RETRY TIME INTERVAL

For scheduled active requests, the time interval is the number of minutes that you want STROBE to wait before trying to initiate the measurement session again, if STROBE initially cannot find the target job. This field appears only if you are using the scheduling function of the STROBE Advanced Session Management Feature.

RETRY INTERVAL COUNT

When you schedule an active request, you can specify the number of times you want STROBE to attempt to initiate the measurement session, if STROBE initially cannot find the target job. The number in this field is the number of times remaining from the number of measurement session initiation attempts that you requested. This field is not shown in Figure 3-2 on page 3-4 and appears only if you are using the scheduling function of the STROBE Advanced Session Management Feature.

REQUEST GROUP

The name of the request group of which this measurement request is an element. A request group is a collection of measurement requests that have been saved together under one name with the STROBE Advanced Session Management Feature. This field appears only if you are using the STROBE Advanced Session Management Feature.

CONCURRENT SET NAME

The name of the concurrent set to which this request group element belongs. A concurrent set consists of two types of request group elements: a trigger request and one or more related requests. The trigger request is denoted with (T). This field appears only if you are using the STROBE Advanced Session Management Feature.

CONCURRENT SET ELEMENTS

The job names of the request group elements that comprise the concurrent set. The trigger request, if present, is denoted with (T), which appears to the left of the trigger request jobname. STROBE lists a maximum of 20 request group elements in this field. If the concurrent set contains more than 20 request group elements, a plus sign (+) appears to the right of the last request group element. This field appears only if you are using the STROBE Advanced Session Management Feature.

SYS REQ

The system(s) on which you selected to run your measurement request when you operate STROBE in a multisystem environment. If you specified that this request can run on any system in the SYSPLEX, this field will contain, "ALL SYSTEMS." If you select a list of systems, the list will appear in this field. If you chose to submit the request on all systems in the SYSPLEX except specified systems, a list of the excluded systems will appear with the label "ALL SYSTEMS EXCEPT."

MODULE MAPPING BASELINE

The value of the BASELINE operand that you specified on the MODULE MAPPING STROBE/ISPF panel. STROBE does not map load modules with a percentage of execution samples less than this value. This field appears only if specified. For more information on module mapping, see Chapter 2 and Chapter 3 of the *STROBE MVS User's Guide* or the *STROBE MVS User's Guide with Advanced Session Management* Chapter 2 and Chapter 4.

BASELINE OVERRIDE

The names of modules for which STROBE obtained control section mapping information regardless of the percentage of execution samples gathered. STROBE

maps these modules even if their execution of samples is less than what you specified for the MODULE MAPPING BASELINE field. This field appears only if specified.

LIBRARY

The names of the libraries specified in which STROBE located module mapping data. This field appears only if specified.

MAPPED SVCS

The SVC numbers for which you requested detailed reporting. Execution and wait activity within the modules invoked by the SVCs appears in subsequent reports in the Performance Profile. This field appears only if specified.

OPTIONS

The attribution or data collector operands that you specified. Review these options to verify those that you specified in your measurement request. For more information on specifying data collector options, see Chapter 2 of the *STROBE MVS User's Guide* or the *STROBE MVS User's Guide with Advanced Session Management*.

Measurement Statistics

The Measurement Statistics section provides the highest-level summary of the system resources used during the measurement session. Verify this section to determine if STROBE collected enough data for you to improve performance and to guide your further investigation.

CPS TIME PERCENT

The percentage of time during which the central processing subsystem (comprising one or more CPUs) was in use by application tasks executing within the measured job step. STROBE computes the value as the ratio of samples in which it observed one or more sites of execution to the total number of samples taken. A high CPS time percent alerts you to focus on these reports:

- Program Section Usage Summary report
- Program Usage by Procedure report
- Attribution of CPU Execution Time reports
- Most Intensively Executed Procedures report
- Subsystem-specific reports that provide CPU consumption information

WAIT TIME PERCENT

The percentage of time during which the central processing subsystem was available but was not in use by application tasks executing within the measured job step. The value is the ratio of wait samples to total samples. A high wait time percent alerts you to focus on these reports:

- Time Distribution of Activity Level report
- Resource Demand Distribution report
- Data Set Characteristics report
- Data Set Characteristics Supplement report
- VSAM LSR Pool Statistics report
- I/O Facility Utilization Summary report
- DASD Usage by Cylinder report
- Attribution of CPU Wait Time report
- Subsystem-specific reports that provide CPU wait time information

RUN MARGIN OF ERROR PCT

The margin of error at a confidence level of 0.95 for the percentages shown in the Performance Profile reports that detail run time. The reports in the Performance Profile that detail run time are reliable within a range of plus or minus the RUN MARGIN OF ERROR PCT reported. A RUN MARGIN OF ERROR PCT of less than 2% usually reflects a reliable measurement. Check that this value is 2% or less to verify that you have collected enough samples to have a valid measurement.

CPU MARGIN OF ERROR PCT

The margin of error at a confidence level of 0.95 for the percentages shown in the Performance Profile reports that detail CPU time. The CPU MARGIN OF ERROR PCT signifies that the reports that detail CPU time are reliable within a range of plus or minus the CPU MARGIN OF ERROR PCT reported. A high CPU margin of error does not mean that the Performance Profile is invalid. Rather, it indicates that you should focus your analysis on the reports that detail run time.

TOTAL SAMPLES TAKEN

The total number of samples collected during the measurement session.

TOTAL SAMPLES PROCESSED

The total number of samples selected for processing. This number may be lower than the value in TOTAL SAMPLES TAKEN. If STROBE collects twice the number of samples specified in TARGET SAMPLE SIZE, STROBE halves its sampling rate. Because STROBE collected more samples while it was sampling at a faster rate, STROBE needs to discard some of the samples collected. STROBE discards enough samples so that the samples processed and the information in the Performance Profile reflect an even distribution of samples over time.

INITIAL SAMPLING RATE

The rate at which STROBE took samples at the beginning of the measurement session, expressed in samples per second to the nearest hundredth. STROBE calculates the initial sampling rate based on the expected session duration and the target sample size (rate=size/duration). STROBE does not allow the sampling rate to exceed 500 samples per second.

FINAL SAMPLING RATE

The rate at which STROBE took samples at the end of the measurement session, expressed in samples per second to the nearest hundredth. If this rate is lower than the initial sampling rate, STROBE has automatically reduced the sampling rate at least once after collecting more than twice the target number of samples. STROBE processes measurements according to the sampling rate in effect at the end of the session. To maintain this distribution, it discards the appropriate number of samples taken at higher sampling rates. These discards do not affect the validity of the measurement, but a final sampling rate lower than the initial sampling rate suggests that you should increase your estimate of the program's run time for later measurement requests. STROBE does not allow the sampling rate to exceed 500 samples per second.

SESSION TIME

The actual duration of the measurement session, expressed in minutes and seconds to the nearest hundredth. If STROBE could not close the sample data set properly (for example, because execution of the job in the address space was cancelled), a warning message appears in place of this field and no values appear in the remaining fields.

Note: If you have suspended and subsequently restarted the measurement session, this field represents the elapsed time from the beginning of the measurement session. The time that the session was suspended is **not** factored out.

CPU TIME

The amount of time during the measurement session that one or more CPUs were executing tasks in the measured job step, exclusive of the measurement task itself, expressed in minutes and seconds to the nearest hundredth. Because you are usually billed for the CPU time used, it is considered the most important measure of system resource time.

WAIT TIME

The estimated portion of run time during which no task within the measured address space was able to make use of the CPU time available to it, expressed in minutes and seconds to the nearest hundredth.

STRETCH TIME

The estimated amount of time that the CPU was unavailable to process programs executing in the measured address space because of demands made by higher-priority address spaces and by service request blocks (SRB) processing time for all address spaces. Stretch time may be high when multiple logical partitions (LPARs) are executing on the same complex of processors.

The value in the STRETCH TIME field reported on Measurement Session Data reports can vary widely, even when you run STROBE several times under what seem to be identical conditions. This discrepancy may result from a variation in system load.

SRB TIME

The time, expressed in minutes and seconds to the nearest hundredth, during which the system was executing Service Request Blocks (SRBs). SRBs represent high-priority service routines. STROBE provides SRB time for your information and does not detail it in subsequent reports. Because STROBE measures while running in task mode, it cannot measure activity running under an SRB. CPU time, which is the aggregate task time for all Task Control Blocks (TCBs) in the address space, does not include SRB time. SRB time that is high relative to CPU time could indicate an unusual or atypical type of processing and may warrant further investigation with other tools.

SERVICE UNITS

The amount of CPU execution activity that STROBE observed during the measurement session, not including the measurement task itself. A service unit is a machine-independent unit of work consisting of a fixed number of instructions.

PAGES IN

The number of page-in operations occurring within the address space during the measurement session.

PAGES OUT

The number of page-out operations occurring within the address space during the measurement session.

PAGING RATE

The average rate, in operations per second, at which page-in and page-out operations were performed for the address space during the measurement session.

EXCPS

The number of EXCP requests (direct invocations of execute channel programs, representing I/O) issued from the measured address space during the measurement session and the average rate at which they were issued. If a timing control table (TCT) was not in use during the measurement session, this value is zero.

A high EXCP rate correlated with high wait time alerts you to look for the causes of excessive I/O activity in the following reports:

- Data Set Characteristics report
- Data Set Characteristics Supplement report

- VSAM LSR Pool Statistics report
- I/O Facility Utilization Summary report
- DASD Usage by Cylinder report

Report Parameters

Verify this section to ensure that you have produced the STROBE Performance Profile with the detail that enables you to conduct your analysis efficiently. The following fields always appear in this section: DASD, DASDGAP, SORTSIZE, LINES/PAGE, and REPORT RESOLUTION. Examine these fields to validate modifications to the expected appearance of reports. The remaining fields appear only if specified. See the *STROBE MVS User's Guide*, Chapter 6, or the *STROBE MVS User's Guide with Advanced Session Management*, Chapter 8, for additional information about these fields.

REPORT RESOLUTION

The specified target size in bytes for the codeblocks into which STROBE collected consecutive instructions for detailed reporting. The default value is 64 bytes. For more information, see "Program Usage by Procedure Report" on page 3-42.

SORTSIZE

The specified sort core size available to the system sort/merge program.

LINES/PAGE

The maximum number of lines to be printed on each page of the Performance Profile. The default value is 60 lines per page.

DETAIL

The system modules or control sections the DETAIL parameter specified. STROBE normally combines all activity within system modules into pseudo-entities, printing a single line for each control section of a system module. For the specified modules, the DETAIL parameter overrides the compression of the specified modules. See the *STROBE MVS User's Guide*, Chapter 6 or the *STROBE MVS User's Guide with Advanced Session Management*, Chapter 8, for a further description of the reporting parameters.

COMPRESS

The module prefixes that were specified to be compressed into pseudo-sections with the COMPRESS option.

DASDGAP

The number of inactive contiguous cylinders that STROBE can pass over while continuing to accrue resource use as reported on the DASD Usage by Cylinder report. A larger value in this field will make the DASD Usage by Cylinder report more concise. The report always shows this parameter. The default value is five.

DASD=*nn.n*%

The baseline specified percentage of DASD usage that a cylinder must exhibit for STROBE to print a separate section in the DASD Usage by Cylinder report. The report always shows this parameter. The default value is 2%.

JAVA TARGETING ENABLED

Indicates that the Java Feature measurement targeting data was specified.

NOMEMOBJ

Indicates that the z/OS Memory Objects Report was suppressed and will not appear

NOPROC

Indicates that the Program Usage by Procedure report was suppressed and will not appear.

NODSCS

Indicates the Data Set Characteristics Supplement Report was suppressed and will not appear.

NODASD

Indicates that the DASD Usage by Cylinder report was suppressed and will not appear.

NOTRAN

Indicates that the Transaction Usage by Control Section report was suppressed and will not appear.

NOTASK

Indicates that the reporting of the execution of all tasks within the program is combined into a single line in the Time Distribution of Activity Level and Resource Demand Distribution reports. If NOTASK does not appear, these reports show a separate line for each task.

ALLCSECT

Indicates that a report line for each inactive control section that STROBE identified will appear in the Program Section Usage Summary report. If you do not specify ALLCSECT, the report does not list inactive control sections.

ALLTASK

Indicates that all tasks will appear on the Resource Demand Distribution report and the Time Distribution of Activity Level report. If you do not specify this parameter, the reports have separate entries for the six most active tasks, and the remaining tasks are grouped together, with their activity reported under OTHER.

ALLDD

Indicates that all ddnames will appear on the Resource Demand Distribution report and the Time Distribution of Activity Level report. If you do not specify this parameter, the reports have separate entries for eleven of the most active ddnames, and the remaining ddnames are grouped together, with their activity reported under OTHER.

MEISA

Indicates that the Performance Profile includes the Most Extensive Inactive Storage Areas report. The report is suppressed by default and does not appear unless specified.

WAITLOC

Specifies that relevant reports show the locations of wait within the program. If you do not specify WAITLOC, the reports provide the total wait time for each program without noting the locations.

PUBP=*nn.n*%

The specified baseline percentage of execution samples a codeblock must exhibit for STROBE to print a separate codeblock line in the Program Usage by Procedure report. The default value is zero.

TUCS=*nn.n*%

The baseline specified percentage of execution samples that a transaction must exhibit for STROBE to print a separate section in the Transaction Usage by Control Section report. The default value is zero.

ATTR=*nn.n*%

The baseline percentage of execution or wait samples that a system service must exhibit for STROBE to print an Attribution report. If the system service has less than

the specified percentage of samples, STROBE does not print the report. The default value is 2%.

ATTRLINE=*nn.n%*

The baseline percentage of execution or wait samples an attributed module must exhibit for STROBE to include it in an Attribution report. If the module has less than the specified baseline percentage for a particular system service, STROBE does not print the detail line for the module in the report for the service. The default value is zero.

DATE FORMAT

The date format specified in the STROBE parameter data set. This format appears on certain STROBE/ISPF panels and Performance Profile reports.

TIME FORMAT

The time format specified in the STROBE parameter data set. This format appears on certain STROBE/ISPF panels and Performance Profile reports.

COLLHIST

Indicates that a measurement session history record was requested for this measurement session when the Performance Profile was produced. You can examine the measurement session history for a specific job step, DBRM, or a transaction using STROBE/ISPF or APMpower. This field appears only if you are using the STROBE Advanced Session Management Feature.

DBRMBASE

The percentage of CPU execution samples a DBRM must exhibit for STROBE to save measurement session information for each DBRM in the history file. If the DBRM has less than the specified baseline percentage, it is not included in the measurement session history. This field appears only if you are using the STROBE Advanced Session Management Feature. The default baseline value is 1%.

TRANBASE

The percentage of CPU execution samples a transaction must exhibit for STROBE to save measurement session information for each transaction in the history file. If the transaction has less than the specified baseline percentage, it is not included in the measurement session history. This field appears only if you are using the STROBE Advanced Session Management Feature. The default baseline value is 1%.

REPJOB

The name specified to replace the job name in the measurement session history record. This field appears only if you are using the STROBE Advanced Session Management Feature.

REPSTP

The name specified to replace the step name in the measurement session history record. This field appears only if you are using the STROBE Advanced Session Management Feature.

Time Distribution of Activity Level Report

The Time Distribution of Activity Level report (Figure 3-3 on page 3-14) displays a chronological record of the activity level of the program or subsystem and of its primary file access activities. This report can help you determine the highest users of CPU time, particularly among storage devices.

Figure 3-3. Time Distribution of Activity Level Report

```
                                   ** TIME DISTRIBUTION OF ACTIVITY LEVEL **

TASK OR   RESOURCE   N X 10 PLUS OR MINUS 5 IS PERCENT OF FULL UTILIZATION   * IS GREATER THAN 95%     - IS LESS THAN 5%
 DDNAME           .-------------------------------------------------------------------------------------------------.
STRBSAM1  CPU      .  13333351934437144451443131451343 5 14354184551333147375 734443533447 1347347111 33343131 314 413  5.
NOBLOCK   3380     .   4454337 34351334414343445145554 3 747314 13444411535145335343 3 53317411144 1131 3 151313113 53 1  .
LOBLOCK   3380     .   11 141113 111 41111 14131 113 4 1311 33 11 311 1  11 3111331 3 1 13313 1   7 31131 5 4151141451  .
.FILEMGT           .4151                                  1                                                        375.
HIBLOCK   3390     .         1      3     1            3     1       14     1       3      3  1      1         1      1 .
STEPLIB   3380     .1                                                                                               .
                   0----0----1----1----2----2----3----3----4----4----5----5----6----6----7----7----8----8----9----*.
                   0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----*.
                   START RUN                 PERCENT OF ALLOCATED RUN TIME                                END RUN
```

TASK OR DDNAME

The first entries under TASK OR DDNAME (data definition name) describe task execution activity within the address space. Subsequent entries describe file access activities. The report lists the activities within each task or ddname in order of decreasing solo activity.

The report limits the number of task entries to six and the number of file access activities to eleven. It defines the top five tasks and groups the rest in .OTHER. Also, if the program or subsystem has more than eleven active files, it groups the least active in an entry labeled .OTHER. If you specify the ALLTASK parameter, the report will show all the tasks. Similarly, if you specify the ALLDD parameter, the report will list all the tasks or ddnames in which STROBE detected activity.

The report identifies tasks by the name of the module in which execution was initiated or by the name specified in the invocation of an ATTACH macro. When more than one task executes the same load module, STROBE prints the module name and a unique identifying number that corresponds to a task control block (TCB) address. For example, at the first occurrence of a load module that is executed in multiple tasks, STROBE prints the module name: LOADMOD 1, with the 1 being the unique identifying number. At the second occurrence of the same load module, STROBE prints: LOADMOD 2. At the third occurrence of the same load module, STROBE prints: LOADMOD 3. The unique identifier differentiates the instances that the load module occurs. If the load module executes only once, STROBE does not associate a number with the load module's name.

If you specified the NOTASK reporting option, this report contains a single line showing the combined time distribution of all tasks.

The separate entry labeled .FILEMGT combines file access operations (such as open and close activities) that STROBE could not assign to a specific file access activity. If their collective activity is low compared to the accompanying tasks (the report limits the number of task entries to six), the report may include the file access operations in the .OTHER line entry instead.

RESOURCE

For file access activities, the model number of the I/O device that serviced each task. If the model is not available, this column can show the following device types:

C	Communication
DA	Direct access
G	Graphic
MT	Magnetic tape
UR	Unit record

For CPU activities, the literal "CPU" appears in this column.

For files processed by the BatchPipes subsystem, the literal “B_PIPES” appears in this column.

Time Distribution Chart

The time distribution section charts the distribution of processing activity on a horizontal scale with 100 graduations. Each vertical segment represents 1% of the measurement session. Consequently, any point on the chart corresponds to the execution of a task or file access activity during a particular one-hundredth of the measurement session, expressed as a proportion of all activity during that segment. STROBE rounds the values and represents them as follows:

- “*” indicates activity greater than 95%.
- A single-digit entry indicates the activity percentage in tens.
- “-” indicates activity less than 5% but greater than zero.
- A blank indicates no activity.

Because file access activities and task execution activities can overlap one another, the percentages in any column can amount to more than 100, depending on the extent of overlap.

Resource Demand Distribution Report

The Resource Demand Distribution report (Figure 3-4 on page 3-15) summarizes the use of CPU and I/O resources in the measured job step. STROBE bases all percentages of resource use on the total run time. The tasks and resources listed are the same as those in the Time Distribution of Activity Level report. The report lists them in order of decreasing solo activity, as indicated by the values shown in the column headed SOLO IN EITHER.

If you specified the NOTASK reporting option, this report contains a single line showing the combined time distribution of all tasks.

Figure 3-4. Resource Demand Distribution Report

** RESOURCE DEMAND DISTRIBUTION **

TASK OR DDNAME	RESOURCE	PERCENT OF RUN TIME SERVICED BY CPU	PERCENT OF RUN TIME SERVICED BY I/O	PERCENT OF RUN TIME SERVICED BY EITHER	PERCENT OF RUN TIME SPENT SOLO IN CPU	PERCENT OF RUN TIME SPENT SOLO IN I/O	PERCENT OF RUN TIME SPENT SOLO IN EITHER	PERCENT OF RUN TIME SPENT CAUSING CPU WAIT	CUMULATIVE PERCENTAGES SOLO TIME	CUMULATIVE PERCENTAGES CAUSING CPU WAIT
STRBSAM1	CPU	30.91	.00	30.91	30.91	.00	30.91	.26	30.91	.26
NOBLOCK	3380	.13	27.87	28.01	.13	27.74	27.87	40.16	58.78	40.42
LOBLOCK	3380	.00	15.06	15.06	.00	15.06	15.06	18.23	73.84	58.65
.FILEMGT		.66	2.11	2.77	.66	2.11	2.77	4.23	76.61	62.88
HIBLOCK	3390	.26	2.38	2.64	.26	2.25	2.51	5.15	79.12	68.03
STEPLIB	3380	.00	.13	.13	.00	.13	.13	.00	79.25	68.03

TASK OR DDNAME

The first entries under TASK OR DDNAME (data definition name) describe task execution activity within the address space. Subsequent entries describe file access activities. The report lists the activities within each task or ddname in order of decreasing solo activity.

The report limits the number of task entries to six and the number of file access activities to eleven. If the program or subsystem has more than six active tasks or eleven active files, the report groups the least active in an entry labeled .OTHER.

If you specify the ALLTASK parameter, then the report will show all of the tasks. Similarly, if you specify the ALLDD parameter, then the report will list all of the ddnames in which STROBE detected activity.

The report identifies tasks by the name of the module in which execution was initiated or by the name specified in the invocation of an ATTACH macro. When more than one task executes the same load module, STROBE prints the module name and a unique identifying number that corresponds to a task control block (TCB) address. For example, at the first occurrence of a load module that is executed by multiple tasks, STROBE prints the module name: LOADMOD 1, with the 1 being the unique identifying number. At the second occurrence of the same load module, STROBE prints: LOADMOD 2. At the third occurrence of the same load module, STROBE prints: LOADMOD 3. The unique identifier differentiates the instances that the load module occurs. If the load module is executed only once, STROBE does not associate a number with the load module's name.

The separate entry labeled .FILEMGT is a pseudo-activity, which combines file access operations (such as open and close activities) that the STROBE could not assign to a specific file. If their collective activity is sufficiently low, the report will include them in the .OTHER line entry instead. (For more information on pseudo-activities, see "Pseudo-Entities" on page 3-2.)

RESOURCE

For file access activities, the model number of the I/O device that serviced each task. If the model is not available, this column can show the following device types:

C	Communication
DA	Direct access
G	Graphic
MT	Magnetic tape
UR	Unit record

For CPU activities, the literal "CPU" appears in this column.

For files processed by the BatchPipes subsystem, the literal "B_PIPES" appears in this column.

SERVICED BY CPU

The percentage of run time spent by a CPU in servicing or performing the activity. The sum of CPU service percentages is equal to or greater than the CPS time percentage reported in the Measurement Session Data report (Figure 3-2 on page 3-4). This percentage may be greater than the CPS percentage in a multitasking environment when several CPUs simultaneously execute for the measured job step.

The reported CPU service time percentage for execution of a task includes the time required for the program itself, the subroutines it invokes, library routines, and some supervisor calls. It does not include CPU time associated with file management operations assigned to individual file access activities or to .FILEMGT.

The CPU service time percentage reported for a file access activity represents the CPU time required to execute IBM-supplied access method routines in accessing the subject data set. STROBE assigns activity in data management SVCs, some ISAM and BDAM access method modules, and all unidentified files to the pseudo-activity .FILEMGT.

Note: A high percentage of run time in a CPU resource indicates that a small percentage of time processing an application program involved I/O. With this information, you can refer to the Program Section Usage Summary report (Figure 3-13 on page 3-39) and analyze a specific application program's resource use.

SERVICED BY I/O

The percentage of run time spent by I/O facilities in servicing the activity. Service time includes all time during which the servicing unit was busy performing the required access operations. For example, service time for disk I/O operations includes time spent by the servicing disk access mechanism during the seek, search, and data transfer operations.

The I/O facility use by each file access activity is further broken down by unit and volume in the I/O Facility Utilization Summary report (Figure 3-10 on page 3-34). For more information on file access activity, consult that report.

For task execution, the I/O service time percentage is always zero. Service time for access to elements such as program load modules and overlays is shown in STEPLIB or JOBLIB activity. STROBE assigns I/O operations performed simultaneously for several data sets (such as multiple open and close operations) to the pseudo-activity .FILEMGT.

SERVICED BY EITHER

The percentage of run time in which either CPU or I/O facilities were servicing the activity.

SOLO IN CPU

The percentage of run time during which a CPU was servicing the activity and no file access activity was occurring.

For task execution activity, CPU solo time is the time during which the task was being executed and no other activity was taking place.

For a file access activity, CPU solo time is the time during which a CPU was executing file management routines for the activity and no CPU or I/O facility was servicing any other activity within the address space.

Note: High CPU solo time represents a performance improvement opportunity. You may be able to restructure the program so that it takes advantage of more than one CPU at one time.

SOLO IN I/O

The percentage of run time spent by the I/O facility in servicing the file access activity, during which no I/O facility or CPU was servicing any other activity within the address space. The I/O Facility Utilization Summary (Figure 3-10 on page 3-34) further subdivides solo I/O facility usage for each file access activity by unit and volume. For a task execution activity, the I/O solo time is always zero. High I/O solo time represents a performance opportunity. See “Report Terminology” on page 3-1 for more information on interpreting this field.

SOLO IN EITHER

The percentage of run time during which one activity alone was being serviced for the measured job step, whether the service was by the CPU, an I/O facility, or both. SOLO IN EITHER represents the sum of:

- CPU solo time percentage for the activity
- I/O solo time percentage for the activity
- percentage of run time during which both CPU and I/O facilities were servicing the activity but no other activity was being serviced

CAUSING CPU WAIT

The percentage of run time during which execution was delayed pending the occurrence of an awaited event, usually an I/O completion event.

For a task execution, CAUSING CPU WAIT represents the time during which execution for the measured job step was blocked, waiting for a completion event not associated with any particular file access. In the case of online subsystems, this time is most commonly simple terminal I/O wait—wait incurred because there are no transactions to process.

Other factors that may contribute to wait time reported in a .FILEMGT execution activity include tape mounts, module loading, inter-region communication, operator replies, and timer requests. If the module in which the wait occurs is not a standard access method module, the report includes in a line entry for a task time spent waiting for completion of any of several file access operations — a multiple wait condition. Normally, this percentage of wait time cannot be reduced.

For file access activities, CAUSING CPU WAIT represents the time during which execution in the measured job step is blocked while waiting for an I/O operation to complete. Normally, this value is greater than or equal to the SOLO IN I/O value for the activity. It includes the time during which the activity was enqueued because of contention in the system for the required I/O facility. It does not include time spent waiting for data sets to be opened. (Such activity is reported in the .FILEMGT line.)

Among multiple waiting tasks, STROBE assigns wait to the task that is suspended because of file I/O.

Time spent waiting in a standard access method module for the occurrence of one of several events — for example, for the completion of OPEN and CLOSE operations — is assigned to the .FILEMGT file access pseudo-activity.

Note: If there is excessive contention from other active address spaces for the required I/O facility, the activity may be delayed while the system enqueues I/O requests before the initiation of I/O operations. The report shows the delay in a much higher CAUSING CPU WAIT value than that for SERVICED BY I/O. In this case, implementing IBM-developed I/O avoidance technologies such as caching, LSR pooling, hiperbatch, and BatchPipes may reduce wait time.

CUMULATIVE PERCENTAGE SOLO TIME

The cumulative total for each of the values in the SOLO IN EITHER column, for line entries down to and including itself.

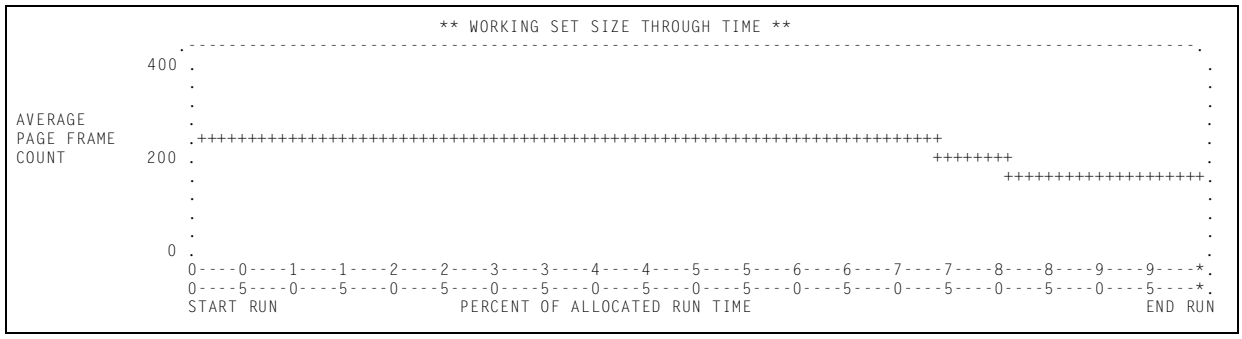
CUMULATIVE PERCENTAGE CAUSING CPU WAIT

The cumulative total for each of the values in the CAUSING CPU WAIT column, for line entries down to and including itself. The last cumulative total equals the WAIT TIME percentage in the Measurement Session Data report.

Working Set Size Through Time Report

The Working Set Size Through Time report (Figure 3-5 on page 3-19) displays a chronological record of the variations in active working set size (user pages that are active) within the address space throughout the measurement session. The report allows you to see how much physical storage was allocated to the measured step and how it changed over time. A very uneven distribution may indicate excessive paging.

Figure 3-5. Working Set Size Through Time Report



Format

The report is in a chart, with the horizontal scale representing time and the vertical scale indicating the average count of page frames. Each of the 100 graduations on the horizontal scale represents 1% of the total measurement session. The position of the "+" on the vertical scale indicates the average number of 4K real storage frames assigned to the address space during each 1% interval of the measurement session. The vertical scale adjusts to accommodate the maximum number of page frames assigned during the measurement session.

Wait Time by Module Report

The Wait Time by Module report (Figure 3-6 on page 3-19) shows all the modules, control sections, pseudo-modules, and pseudo-sections in which STROBE found the target program in the wait state. If you specify the WAITLOC parameter when you create the Performance Profile, the OFFSET column shows the location of wait within each module for both paging and total wait activities, and the COMPRESSED SECTION column is suppressed. In the case of multiple tasks waiting, STROBE selects the task most likely to be delaying the step.

Figure 3-6. Wait Time by Module Report

```
                                        ** WAIT TIME BY MODULE **

MODULE   SECTION COMPRESSED   FUNCTION                RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:   .78%
 NAME      NAME    SECTION                             PAGE     TOTAL        .00     23.00     46.00     69.00     92.00

.DB2     DSNSLD1           STORAGE MANAGEMENT DRIVR     .00       .01        .
.DB2     DSNVSR            SUSP/RES/CANCEL SYNCHRON     .00       .27        .
.DB2     DSNXGRDS DSNXGSFL BLD SFL STRCTR, SET FUNC     .00       .01        .
.DB2     DSNXGRDS DSNXGT2J SETUP RTAT2J, TYP 2 JOIN     .00       .01        .
.DB2     DSNXGRDS DSNXOMJS FND JOIN PREDS, MTCOL MJ     .00       .01        .
.DB2     DSNXGRDS DSNXOSR  ASL GENERATR CREATE SORT     .00       .02        .
.DB2     DSNXGRDS DSNXOSSP RDS ACCESS MODULE GENER      .00       .01        .
                                                       -----     -----
.DB2     TOTALS            DB2 SYSTEM SERVICES          .00       .34
```

MODULE NAME

The name of the load module or pseudo-module in which STROBE found the target program in the wait state. For module names that exceed eight characters, see the note in the SECTION NAME paragraph.

SECTION NAME

The name of the control section or pseudo-section in the load module or pseudo-module identified by MODULE NAME.

Note: For all module or section names that exceed eight characters, STROBE generates a *token*, which is an eight-byte identifier. The token comprises the first four characters of the module or section name followed by a hyphen (-) and then the last three characters of the name. Refer to the Token - Longname Cross Reference report, Figure 3-22 on page 3-52, to reconcile all tokens with their long names.

COMPRESSED SECTION

The name of the control section of the module that has been compressed within a pseudo-module, such as .DB2. STROBE groups supervisory functions of the operating system in pseudo-modules. For more information on pseudo-modules, see "Pseudo-Entities" on page 3-2.

FUNCTION

A brief description of the module's function, employing the STROBE descriptions for SVCs and commonly used operating system and subsystem modules. The field also displays any descriptions that the STROBE system programmer has provided for control sections and modules.

OFFSET

The hexadecimal location within the control section at which the module entered the wait state, and appears only if you specified the WAITLOC parameter.

RUN TIME PERCENT

The percentage of time during the measurement session that the address space was in the wait state. There are two measures of run time:

- PAGE shows wait time that results from retrieving a page from the page data set. A high value in this column indicates that there is not enough physical memory assigned to the address space. If you noticed a high paging rate on the Measurement Session Data report, the Wait Time by Module report enables you to see which module was experiencing delay because of paging.
- TOTAL measures all causes of wait time, including page retrieval, programmed I/O operations, and timer requests.

RUN TIME HISTOGRAM

Represents the distribution of wait time. The asterisk symbol "*" indicates page wait. The plus symbol "+" indicates other wait time.

MARGIN OF ERROR

For run-time percentages, this is the same value that appears in the RUN MARGIN OF ERROR PCT field of the Measurement Session Data report. (See "RUN MARGIN OF ERROR PCT" on page 3-9.)

Data Set Characteristics Report

The Data Set Characteristics report (Figure 3-7 on page 3-21) shows the characteristics of all data sets accessed for the measured job step during the measurement session. It lists entries alphabetically by the ddname and order of concatenation.

Both the Data Set Characteristics report and the Data Set Characteristics Supplement include an entry for a data set if STROBE detected activity in the data set or computed a change in EXCP counts for it. If the EXCP count changed but STROBE did not detect activity in the data set, the report shows only the ddname and the EXCP count, and other I/O reports in the Performance Profile have no entry for it.

STROBE records the characteristics from the Data Control Block (DCB), the Job File Control Block (JFCB), or the Access Control Block (ACB) the first time it detects activity in the data set. It does not note whether subsequent activity has different characteristics.

Figure 3-7. Data Set Characteristics Report

```
                                        ** DATA SET CHARACTERISTICS **

DDNAME     ACCESS    POOL  REC  BLK/CI    HBUF  BUF  RPL  -SPLITS-      EXCP DATA SET NAME
           METHOD     NO   SIZE  SIZE       NO   NO STRNO CI    CA    COUNTS

HIBLOCK    QSAM             200  6400            5                       63 SYS93271.T144729.RA000.WPAAZSSW.R0004148
LOBLOCK    QSAM             200  1000            5                      400 SYS93271.T144729.RA000.WPAAZSSW.R0004149
NOBLOCK    QSAM             200   200            5                     2000 SYS93271.T144729.RA000.WPAAZSSW.R0004150
STEPLIB    BPAM                 32760                                    3 WPA.APF.LOADLIB
PGM=*.DD
```

DDNAME

The ddname assigned to the data set. A program may access a VSAM data set by more than one ddname, but STROBE tracks the VSAM data set by only one ddname. For VSAM files in which data set name sharing is in effect, the ddname that STROBE tracks depends upon the information available to it when the file is first active. You can access VSAM data sets through other ddnames such as the ddname assigned to the base cluster or path. For alternate indexes in the upgrade mode, STROBE creates a ddname by appending a number to the ddname associated with the base cluster of the alternate index. STROBE reports PGM=*.DD in this column to indicate the use of a compile/link/go procedure. A refer-back results in a DDNAME entry of PGM=*.DD in the Task Input/Output Table.

For Hierarchical File System (HFS) and zSeries File System files that are allocated to the measured MVS task, the ddname will be the MVS ddname. For all other HFS files, the ddname will be ".HFSnnnn".

ACCESS METHOD

The access method used to process the data set:

B_PIPES	Files processed by the BatchPipes subsystem
BDAM	Basic direct access method
BDAM FP	Basic direct access method fast path
BPAM	Basic partitioned access method
BSAM	Basic sequential access method
EXCP	Direct invocation of execute channel program
HFS	Hierarchical File System
IAM	Accessing an IAM* file
IAMLOAD	Loading an IAM* file
	*IAM refers to Innovation Access Method, a product of Innovation Data Processing, Inc.
ISAM	Indexed sequential access method
MEDIA MGR	Media Manager
MM/EXTSEQ	Media Manager on behalf of extended sequential format data sets
MM/PDSE	Media Manager on behalf of partitioned data set extended
OSAM	Overflow sequential access method
OTHER	Nonstandard access method

QSAM	Queued sequential access method
TCAM LINE	Telecommunications access method line group
TCAM MSG	Telecommunications access method message queue
VIO	Virtual input /output
VSAM	Virtual storage access method
zFS	zSeries File System

For VSAM, this field may contain:

VSAM AIXD	Data portion of an alternate index
VSAM AIXX	Index portion of an alternate index
VSAM ESDS	Entry sequenced data set
VSAM KSDS	Key sequenced data set
VSAM INDX	Index of a key sequenced data set
VSAM LDS	Linear data set
VSAM RRDS	Relative record data set
VSAM UPGD	Data portion of an alternate index in the upgrade mode
VSAM UPGX	Index of an alternate index in the upgrade mode
VSAM VRRD	Variable length relative record data set

The VSAM values may be followed by:

FP	Fast path
GSR	Global shared resource
LSR	Local shared resource

If a file is processed without shared resources and as an LSR, it will be identified as an LSR.

POOL NO

The LSR pool number to which the data set is assigned. A value in this field appears only if LSR buffering is in effect. “VSAM LSR Pool Statistics Report” on page 3-32, shown in Figure 3-9 on page 3-32, supplies additional information on pool activity.

REC SIZE

The size of the maximum length record in bytes.

BLK/CI SIZE

The block size of the data set, or VSAM control interval size, in bytes. If the data set is a concatenation of data sets, it gives the block size of the first data set in the concatenation.

Note: For files with high wait percentages in the Resource Demand Distribution report, the blocking factor may cause the inefficiency. See the Data Set Characteristics report to review the blocking factor.

HBUF NO

The number of hiperspace buffers assigned to the LSR pool in which the file participates, not to the individual data set. The report shows the values only for VSAM files.

BUF NO

A count of buffers, depending upon the specified access methods and buffering methods.

- For nonhiperspace buffers that QSAM, IAMLOAD, or VSAM access methods manage, BUF NO shows the number of buffers assigned to the data set.
- When LSR buffering is in effect, BUF NO shows the number assigned to the LSR subpool, not to the individual data set.
- For IAM data sets, BUF NO is the highest buffer number assigned to the data set that STROBE detected during measurement. This value is because IAM dynamically manages buffers.

The BUF NO field does not show a value for applications that manage their own buffers, such as CICS temporary storage management, any of the Basic Access Methods (BDAM, BPAM, and BSAM), and the Overflow Sequential Access Method (OSAM).

Note: Normally, QSAM and IAM access methods manage buffers efficiently. VSAM, however, often requires additional considerations in buffer allocation.

RPL STRNO

For VSAM data sets, the number of request parameter lists that may be processed concurrently. When LSR is active for a file, the value reported pertains to the LSR subpool, not to the individual data set.

CI SPLITS

For VSAM data sets, the number of control interval splits that occurred during the measurement session. CI splits are not reported for records added to a key sequenced data set (KSDS) at end of file (EOF).

Note: A high number of CI splits indicates that you should reorganize the file or allocate more free space.

CA SPLITS

For VSAM data sets, the number of control area splits that occurred during the measurement session.

Note: A high number of CA splits indicates that you should reorganize the file or allocate more free space. CA splits are not reported for records added to a KSDS at EOF.

EXCP COUNTS

For VSAM data sets, the number of physical I/O operations performed on the data set during the measurement session. Because STROBE derives VSAM counts from the samples taken during the measurement session, they may not match the EXCP counts recorded by SMF.

Note: With ICI data sets, STROBE does not show a value in this field. This omission occurs because VSAM does not keep the necessary statistics.

For non-VSAM data sets, this value represents the number of blocks of data transferred during the measurement session, not the number of physical I/O operations performed on the data set.

For concatenated data sets on the same volume, EXCP counts appear on the entry for the first data set within the concatenation.

For data sets in the Hierarchical File System (HFS), EXCP counts are collected only if the STROBE UNIX System Services Feature is installed at your site.

Note: When the number of EXCP counts is high, verify that the block size is optimal to ensure the least I/O operations for the block size with the most data stored upon the device.

DATA SET NAME

The name of the data set associated with the ddname. The data set name reported reflects the information available to STROBE when the file is first active. For a VSAM KSDS, the data set name may be a cluster name or a component name. For a VSAM alternate index, the data set name may be the component name, path name, or cluster name. For HFS files, an entry of "*PATHNAME" indicates that the full pathname as it appears on the SMF record appears in the DSNAME field on the Data Set Characteristics Supplement report. This entry appears when the pathname length exceeds 44 characters.

For information on data set activity, review this and other reports that give data set information in addition to DDNAME information, such as the Data Set Characteristics Supplement report.

Data Set Characteristics Supplement Report

The Data Set Characteristics Supplement report (Figure 3-8 on page 3-25) provides additional information on data sets. This report includes an entry for a data set if STROBE detected activity in the data set or computed a change in EXCP counts for it. If the EXCP count changed but STROBE did not detect activity in the data set, the report shows only the ddname and the EXCP count, and other I/O reports in the Performance Profile have no entry for the data set.

You can cross-reference this report with the Data Set Characteristics report by data definition name, access method, and data set name. If no information exists, STROBE suppresses the names of the fields. You can suppress the report by supplying the NODSCS parameter.

The report provides supplementary information in five sections:

- identifying information
- file creation data
- System Managed Storage (SMS) data (if SMS manages the data set)
- VSAM data (if the data set was created through VSAM)
- other information

Figure 3-8. Data Set Characteristics Supplement Report

```
                                ** DATA SET CHARACTERISTICS SUPPLEMENT **

DDNAME   ACCESS     DSNAME                                  OPEN INTENT\PROCESSING MODE
         METHOD


STDCLSR7 VSAM KSDS  WPAV.WPACI3XX.STDCLSR7                  OUTPUT,DIR
  FILE:  RECFM........                                VSAM:  FREESPACE...  39K    USER RECORDS...          99
         EXTENTS......  1                                    SHROPTS....(2,3)     LOGICAL OPERATIONS:
                                                             CI/CA......   22      DELETES........           1
         EXTNDED FMT.. EA(XRBA),COMP 16%                     %CI FREE.....  0      UPDATES........           0
                                                             %CA FREE.....  0      RETRIEVES......           1
                                                                                   INSERTS........          98
                                                             ATTR...SPEED


STDCLSR8 VSAM INDEX WPAV.WPACI3XX.STDCLSR8                  OUTPUT,DIR
  FILE:  RECFM........                                VSAM:  FREESPACE...  65K    USER RECORDS...           3
         EXTENTS......  1                                    SHROPTS....(2,3)     LOGICAL OPERATIONS:
                                                             CI/CA......   23      DELETES........           0
                                                             %CI FREE.....  0      UPDATES........          34
                                                             %CA FREE.....  0      RETRIEVES......           0
                                                             INDEX LVLS.    2      INSERTS........           0



SYSUT1   QSAM       EDP.MANAGED.TESTMW                      INPUT
  FILE:  RECFM........  F    SMS:  DATA CLASS........
         EXTENTS......  1          MANAGEMENT CLASS..
                                   STORAGE CLASS.....CACHEMW
                                     DIR MSR/BIAS.....025/WRITE
                                     SEQ MSR/BIAS.....030/WRITE
  STATS: RESP....056.5  IOSQ....056.5
         PEND....000.0  DISC....000.0

SYSUT2   B_PIPES    EDP.BPIPE.FILE1                         OUTPUT,SEQ
  FILE:  RECFM........F          WAIT COUNT ...........000001      PIPE PARTNER(S):
         SUBSYSTEM... BP01       WAIT TIME ...........0000.55      JOB: EDPBPR2   DDNAME: SYSUT1
                                 WAIT-FOR-OPEN-TIME ..0000.00      JOB: EDPBPR3   DDNAME: SYSUT1
                                                                   JOB: EDPBPR5   DDNAME: SYSUT1
                                                                   JOB: EDPBPR1   DDNAME: SYSUT1
                                                                   ADDITIONAL PIPE PARTNERS EXIST
```

Identifying Information

This section of the report supplies the following information about the data sets.

DDNAME

The data definition name as it appears in the Data Set Characteristics report.

ACCESS METHOD

The access method as it appears in the Data Set Characteristics report.

DSNAME

The data set name as it appears in the Data Set Characteristics report. For HFS files that show "*PATHNAME" on the DATA SET NAME field of the Data Set Characteristics report, this field displays the full pathname as it appears on the SMF record.

OPEN INTENT\PROCESSING MODE

Describes parameters the program supplied when it opened the data set. This field can help you to understand better how the measured program accesses or processes the data set.

The non-VSAM values are:

EXTEND — The data set is treated as an OUTPUT data set, but records are added to the end of the data set regardless of what was specified on the DISP parameter of the dd statement.

INOUT — The program accesses this data set first for input and later, without reopening the file, for output. The data set is processed as INPUT if it is a SYSIN data set or a PDSE or LABEL=(,,IN) is specified in the dd statement.

INPUT — The data set is opened for INPUT.

OUTIN — The program accesses this data set first for output and later, without reopening the file, for input.

OUTINX — This data set is treated as an OUTIN data set, except that records are added to the end of the data set regardless of what was specified on the DISP parameter of the dd statement. For PDSEs, OUTINX is equivalent to OUTPUT.

OUTPUT — The data set is opened for OUTPUT. (For BDAM, OUTPUT is equivalent to UPDAT.)

RDBACK — This data set is an input data set positioned to read backward. This option is not allowed for DASD data sets; it is supported only for magnetic tape.

UPDAT — This data set is to be updated in place or, for BDAM, blocks are to be updated or added.

The VSAM values are:

DIR — This data set is accessed with direct access.

INPUT — This data set is used for retrieval of records (the ACB specifies the "IN" MACRF option).

DYN — This data set's index component is processed randomly, which most frequently indicates that the COBOL program specified "ACCESS IS DYNAMIC".

For programs written in languages other than COBOL, the DYN parameter indicates that STROBE observed a VSAM POINT request.

I-O — This data set is for input and output access (the ACB specifies both "IN" and "OUT" MACRF value).

LOAD — Records are written to the data set in load mode.

OUTPUT — This data set is used for storage of new records (the ACB specifies the "OUT" MACRF value.) VSAM access allows both input and output operations against the file.

SEQ — This data set is accessed with sequential access.

SKP — This data set is accessed with skip-sequential access.

FILE

Identifies some basic data set characteristics.

RECFM

This field provides and displays record and block format specifications. The record formats are:

F — Fixed length

U Undefined

V Variable length

The block formats are:

B Blocked

S Spanned (variable length) or standard (fixed length)

For print data sets, the following codes apply:

A ANSI control characters

M Machine control characters

EXTENTS

Displays the maximum number of extents allocated to the data set. STROBE reports on extents for concatenated data sets in this way:

- For concatenated sequential data sets, STROBE reports on extents for individual data sets in the concatenation. STROBE reports these extents only if it detects I/O activity for the data set.
- For concatenated partitioned data sets, STROBE reports extents for individual data sets in the concatenation.

EXTNDED FMT

Displays the presence of DFSMS extended format data sets and the attributes assigned to the data sets. The following values may appear in this field:

EA The data set is defined with the extended addressability attribute. If it is followed by XRBA, this label indicates that the data set was accessed with the XRBA option.

COMP The data set is eligible for SMS compression. It may be followed by:

- %nn, which indicates the amount of compression for compressed data
- ??, which indicates that the compressed size is invalid
- Blank (no percentage is displayed), which indicates that the data is allocated in compressed format; however, the contents of the data are not compressed because either SMS determined that compression would cause excessive overhead or SMS could not select a suitable dictionary.

Note: If the COMP field does not display, SMS may have rejected the request for compression at allocation time because the data set is not sufficiently large to take advantage of the compression attribute.

OTHER

Identifies various file characteristics, such as CNV (control interval processing), ICI (improved control interval processing), UBF (user buffering), VLF (virtual lookaside facility), and VVDS (VSAM Volume Data Set).

SMS Data

System Managed Storage (SMS) data for the data set. The SMS option enables you to identify ways to reclassify a data set to improve its performance.

For VSAM, STROBE may not always provide SMS information. For example, if STROBE could not determine the data set's cluster name, STROBE cannot provide SMS information. If only the component name is available and the component name has a last qualifier .INDEX, .I, .DATA, or, .D, STROBE assumes that the file's cluster name is the component name without the last qualifier and attempts to find information for the cluster name.

STROBE does not collect SMS information for alternate indexes. SMS will assign an alternate index the same management class and storage class as its associated base cluster.

DATA CLASS

The data class to which SMS assigns a data set. DATA CLASS contains a collection of allocation and space attributes for a group of data sets. The storage administrator defines the attributes.

MANAGEMENT CLASS

The management class to which a data set is assigned. MANAGEMENT CLASS displays a collection of management attributes that control:

- the release of allocated but unused space
- the retention, migration, and backup of data sets
- the retention and backup of aggregate groups
- the retention, backup, and class transition of objects

The storage administrator defines these attributes.

STORAGE CLASS

The storage class to which a data set is assigned. STORAGE CLASS provides a named list of storage attributes that identify performance goals and availability requirements as defined by the storage administrator. It provides the criteria that SMS uses to determine an appropriate location for a data set or object.

MSR

Identifies the millisecond response (MSR) objective defined for this data set. MSR is defined for both direct and sequential processing modes and is reported as a three-digit value from 000 through 999.

Must Cache data sets are identified by MSR values that cannot be met without the use of cache. May Cache data sets are identified by a value of 000 (indicating that no MSR was specified) or a value that can be met without the use of cache. Never Cache data sets are identified by a value of 999.

Note: Compare this value with the value identified in the RESP field (average response time) to see if the data set is caching as you would expect, given the specified MSR.

BIAS

Identifies the BIAS value defined for the data set. BIAS is defined for both direct and sequential processing modes. The possible values for this field are READ, WRITE, or NONE.

STATS

Identifies some response-time statistics for data sets. These response-time statistics only appear if you are running DFSMS Release 1.1 or higher.

RESP

Displays the average I/O response time. Examining this value, in conjunction with the value reported in the MSR field provides a method of determining the cache

usage of the data set. Generally, an average response time lower than 25 indicates that the data set was cached.

IOSQ

Displays the average I/O Supervisor queue time. This value represents the average time spent waiting for service during an I/O operation while the device was busy with an operation from another system.

PEND

Displays the average pending time. This value represents the average amount of time spent waiting to connect to a device.

DISC

Displays the average disconnect time. This value represents the average time that the device on which the data set resides was logically disconnected from the channel subsystem. A low average disconnect time usually indicates that the data set has a good hit ratio.

Note: High values in the IOSQ, PEND, and DISC fields may indicate that other data sets on this volume or 3390 string are caching poorly.

VSAM Data

This section of the report appears if STROBE determines that the data set uses the Virtual Storage Access Method (VSAM) and can access the catalog information.

FREESPACE

The number of bytes of free space remaining in the component. When displayed in units of 'K' (Kilobytes), 'M' (Megabytes), and 'G' (Gigabytes), this value will not be exact.

SHROPTS

The share options specified for the VSAM data set. This information may not be available for VSAM data sets processed under earlier versions of the Data Facilities Product (DFP).

CI/CA

The number of control intervals in a control area.

%CI FREE

The percentage of control interval set aside as free space when the cluster is initially loaded, during a mass insert, and after any split of control intervals. %CI FREE applies only to key-sequenced clusters and relative-record clusters with variable record lengths.

%CA FREE

The percentage of each control area set aside as free space when the cluster is initially loaded, during a mass insert, and after any split of control intervals. %CA FREE applies only to key-sequenced clusters and relative-record clusters with variable record lengths.

INDEX LVLS

The number of levels in the index component at the last time STROBE detected activity in the data set.

ATTR

The attributes of the VSAM definition parameter. The possible values are:

RCVY The cluster was defined with the RECOVERY definition parameter, which causes VSAM to preformat each control area before data is written.

SPEED The cluster was defined with the SPEED attribute, which instructs VSAM not to preformat control areas.

IMBD The cluster was defined with the IMBED parameter, which specifies that the sequence-set record for each control area is written as many times as it will fit on the first track of the data control area.

REPL The cluster was defined with the REPLICATE parameter, which specifies that each index record is written on its own track as many times as it will fit.

USER RECORDS

The total number of logical records in the data set the last time STROBE detected activity in the data set. If the total number of user records exceeds 999,999,999, the report displays the amount in units of G (billions of records).

LOGICAL OPERATIONS

The total number of the following types of VSAM operations.

- DELETES is the number of logical records that have been deleted during measurement.
- UPDATES is the number of logical records that have been updated and rewritten to the component during measurement.
- RETRIEVES is the number of logical records that have been retrieved from the component. This count includes "get for update" requests that occur via the REWRITE verb and the GET macro that specifies "for UPD".
- INSERTS is the number of logical records that have been inserted during measurement. This count does not include ESDS or KSDS records that are loaded or added to the end of the data set. It also does not include records inserted during create (load) processing.

The report will not contain LOGICAL OPERATIONS values if the Performance Profile includes the CICS Performance Supplement.

BatchPipes Data

This section of the report appears if STROBE determines that a data set was processed by the BatchPipes subsystem.

Note: For data sets processed by the BatchPipes subsystem, the BUFNO column represents the number of blocks the pipe is capable of holding (the pipe depth).

WAIT COUNT

Displays the number of times the job step was forced to wait because the pipe was either full (for a writer job) or empty (for a reader job).

WAIT TIME

Displays the total amount of time spent waiting on a full pipe (for a writer job) or an empty pipe (for a reader job).

Note: A high value in this field may indicate a processing disparity between one or more of the pipe partners. For example, a reader job is reading the records

from the pipe faster than the writer job is writing them. If this is the case, consider exploring ways to balance the pipeline. For more information, see *IBM BatchPipes/MVS User's Guide and Reference*.

WAIT-FOR-OPEN TIME

Displays the total amount of time spent waiting for a pipe partner (either a writer job or a reader job) to become active. If the value for this field is zero, this field does not appear.

PIPE PARTNER

Displays the names of the jobs that have been designated as reader or writer partners for the measured job. The system name also appears if the pipe partner executed on another system in the BatchPipePlex (SmartBatch environments only).

This field displays a maximum of four pipe partners. The line ADDITIONAL PIPE PARTNERS EXIST appears if there are more than four pipe partners observed during the measurement session.

SUBSYSTEM

Displays the system name, if the pipe partner executed on another system in the BatchPipePlex (SmartBatch environments only).

HFS/zFS Data

This section of the report appears if STROBE determines that the data set accessed is in the Hierarchical File System (HFS) or zFS File System. This section of the Data Set Characteristics Supplement report appears only if the STROBE UNIX System Services Feature is installed at your site.

INODE

Displays the file identifier.

DEVICE NUMBER

An identifier for the mounted file system. When a non-unique INODE is encountered within a measurement, the device number can be used further to identify the file. (An INODE is guaranteed to be unique only within an HFS.)

I/O BLOCKS

READ	The number of blocks read from the file during the measurement.
WRITTEN	The number of blocks written to the file during the measurement.

FILE TYPE

The definition of the file. This field can have one of two values: REGFILE for a normal file or FIFO for a named pipe.

BYTES

READ	The number of bytes read from the file during STROBE measurement.
WRITTEN	The number of bytes written to the file during STROBE measurement.

STROBE reports the I/O blocks and bytes statistics only if System Management Facility (SMF) is configured to produce type 92 records. For more information, see the STROBE *STROBE MVS System Programmer's Guide*.

VSAM LSR Pool Statistics Report

The VSAM Local Shared Resources (LSR) Pool Statistics report (Figure 3-9) shows information about buffer pools and is organized by LSR pool. Each entry line of the report represents a subpool. The components of a VSAM cluster will use the buffer subpool with buffers that are at least as large as the component's CI size. For example, if the data cluster of a file in the Data Set Characteristics report appears as being part of pool number 1 and it has a CI size of 4096, its activity would be reported in the fourth line of Figure 3-9 because that subpool has a buffer size of 4096.

Figure 3-9. VSAM LSR Pool Statistics Report

```
                                   ** VSAM LSR POOL STATISTICS **

POOL  TYPE  STR KEY   BUF BUFNO   HBUFNO  --- RETRIEVES ---  ----- WRITES -----    ------------- HIPERSPACE -------------
  NO         NO LEN   LEN                    WITH    WITHOUT     USER   NONUSER        - READS -          - WRITES -
                                             I/O     I/O                          SUCCESSFUL FAILING  SUCCESSFUL FAILING
   1  DATA   37 255   512    30        0         0         0        0        0           0        0          0        0
   1  DATA   37 255  1024     4        0         0         0        0        0           0        0          0        0
   1  DATA   37 255  2048     3        0         0         0        0        0           0        0          0        0
   1  DATA   37 255  4096    70        0       908     30751       16        0           0        0          0        0
   1  DATA   37 255  8192    15        0        11        38        0        0           0        0          0        0
   1  DATA   37 255 12288    10        0         0         0        0        0           0        0          0        0
   1  DATA   37 255 24576     4        0         0         0        0        0           0        0          0        0
   1  INDEX     255   512    15        0         0         0        0        0           0        0          0        0
   1  INDEX     255  1024     5        0         0         0        0        0           0        0          0        0
   1  INDEX     255  2048    50        0         4     63189        0        0           0        0          0        0
   1  INDEX     255  4096    25        0         7       134        0        0           0        0          0        0
   1  INDEX     255  8192     5        0         0         0        0        0           0        0          0        0
   1  INDEX     255 12288     5        0         0         0        0        0           0        0          0        0
POOL    1 STRING WAITS       MAXIMUM CONCURRENT        19       TOTAL        54
```

POOL NO

The identification number of the pool.

TYPE

The VSAM component accessed. These values identify the resource pool:

INDEX	An index component resource pool.
DATA	A data component resource pool.
ANY	Both data and index components may use this resource pool.

STR NO

The number of placeholders defined for the pool. A string is a request to a VSAM data set requiring positioning within the data set. For each concurrent file operation, VSAM requires one or more strings.

KEY LEN

The maximum key length for data sets in the pool. If none of the data sets is keyed, STROBE displays "0".

BUF LEN

The size of the virtual and hiperspace buffers in the subpool.

BUFNO

The number of nonhiperspace buffers in the subpool.

HBUFNO

The number of 4K hiperspace buffers in the subpool.

RETRIEVES WITH I/O

The number of I/Os that occurred to bring data into the subpool.

RETRIEVES WITHOUT I/O

The number of retrievals during which the requested data was found in the subpool (also known as lookasides).

WRITES USER

The number of writes that the user initiated.

WRITES NON-USER

The number of writes that occurred because no buffer was available. The program did not initiate these writes.

HIPERSPACE READS SUCCESSFUL

The number of times a hiperspace buffer was successfully read from the LSR hiperspace buffer pool to a local LSR buffer pool.

HIPERSPACE READS FAILING

The number of times a hiperspace buffer was reclaimed and could not be read from the LSR hiperspace buffer pool to a local LSR buffer pool.

HIPERSPACE WRITES SUCCESSFUL

The number of times a hiperspace buffer was successfully written from a local LSR buffer pool to a hiperspace buffer pool.

HIPERSPACE WRITES FAILING

The number of times a hiperspace buffer was reclaimed and could not be written from a local LSR pool to a hiperspace buffer pool.

Note: **Hiperspace buffering** *helps to reduce DASD I/O by using expanded storage for LSR buffers. Hiperspace storage is not byte-addressable. Therefore, to refer to data stored in a hiperspace buffer, VSAM must first move one 4K page at a time to a nonhiperspace LSR buffer pool. The* STROBE *reporting of hiperspace statistics represents movement of buffers to and from hiperspace buffer pools.*

Pool *n* STRING WAITS

If available, the set of string wait statistics for each LSR pool number (identified by *n*). The statistics only appear for measurement requests that specify the CICS data collection option "Produce Performance Supplement". For more information on these statistics, see the STROBE *STROBE CICS Feature*.

MAXIMUM CONCURRENT	The highest number of concurrent I/O requests observed during the measurement session that were waiting for an LSR pool string to become available.
TOTAL	The total number of I/O requests observed during the measurement session that were waiting for an LSR pool string to become available.

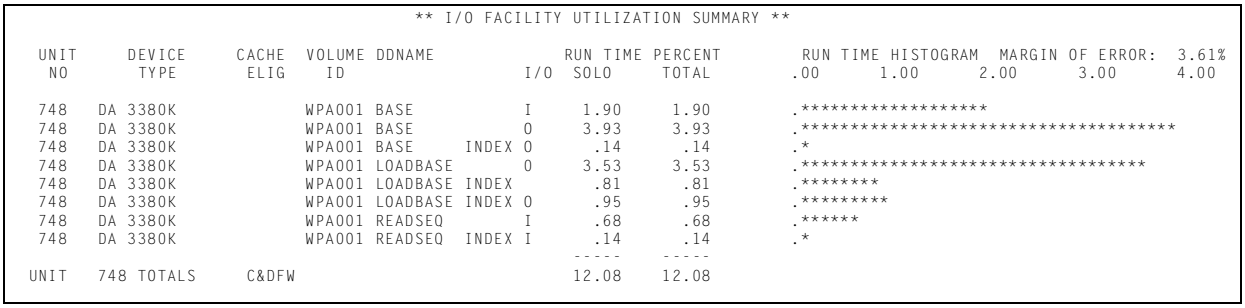
Note: If either of these values is high, consider increasing the number of strings assigned to the pool.

I/O Facility Utilization Summary Report

The I/O Facility Utilization Summary report (Figure 3-10 on page 3-34) shows, by device and volume, the percentage of run time used accessing each data set. The report identifies devices by unit number and device type, volumes by serial number, and data

sets by ddname. It shows index and data components of VSAM data sets separately. This report helps you to analyze file access activity.

Figure 3-10. I/O Facility Utilization Summary Report



```
                                  ** I/O FACILITY UTILIZATION SUMMARY **

 UNIT     DEVICE     CACHE  VOLUME DDNAME             RUN TIME PERCENT        RUN TIME HISTOGRAM  MARGIN OF ERROR:  3.61%
  NO       TYPE      ELIG    ID                 I/O  SOLO     TOTAL        .00      1.00      2.00      3.00      4.00

 748   DA 3380K             WPA001 BASE           I     1.90     1.90        .******************
 748   DA 3380K             WPA001 BASE           O     3.93     3.93        .****************************************
 748   DA 3380K             WPA001 BASE     INDEX O      .14      .14        .*
 748   DA 3380K             WPA001 LOADBASE       O     3.53     3.53        .**********************************
 748   DA 3380K             WPA001 LOADBASE INDEX        .81      .81        .********
 748   DA 3380K             WPA001 LOADBASE INDEX O      .95      .95        .*********
 748   DA 3380K             WPA001 READSEQ        I      .68      .68        .******
 748   DA 3380K             WPA001 READSEQ  INDEX I      .14      .14        .*
                                                        -----    -----
UNIT   748 TOTALS     C&DFW                          12.08    12.08
```

For each device and volume, STROBE combines open and close operations under the pseudo-activity identifier .FILEMGT. Entries are ordered by unit number, volume identifier, and ddname. The report gives subtotals for each unit that has more than one entry.

UNIT NO

The address of the device accessed by the application.

DEVICE TYPE

The type of device and for DASD devices, its model number.

C	Communication
DA	Direct access
G	Graphic
MT	Magnetic tape
UR	Unit record

CACHE ELIG

The cache status of a direct access volume. This field may have the following values:

Blank	The device is not attached to a cache control unit.
ON	The device is being cached; however, neither DASD fast write nor cache fast write is enabled.
OFF	The device is attached to a controller that has cache facilities, but the device is not being cached.
CFW	Cache and cache fast write are enabled for the device; DASD fast write is not enabled on the controller.
DFW	Cache and DASD fast write are enabled for the device; cache fast write is not enabled on the controller.
C&DFW	Cache, cache fast write, and DASD fast write are enabled for the controller.

Note: If this field is blank, showing that a device is not cache eligible, and this device contains a data set whose performance you would like to improve, move the data set to a cache-eligible device.

The CACHE ELIG field does not indicate the cache mode for individual data sets on the volume. STROBE is unable to detect caching status at the data set level because, for SMS data sets, caching is typically managed dynamically. STROBE is also unable to detect cache eligibility for non-IBM caching products that do not emulate IBM caching devices. In such cases, the CACHE ELIG field appears blank.

VOLUME ID

The identifying number of the volume.

DDNAME

Identifies the data definition name.

I/O

Input or output activity. The percentage of time during which STROBE detects input activity is identified with an "I", while the percentage of time STROBE detects output activity is identified with an "O". Entries are ordered by unit number, volume identifier, and DDNAME. Subtotals are given for each unit that has more than one entry.

Note: For DB2 table space files accessed through Media Manager, the I/O entries for this column may be blank.

RUN TIME PERCENT

The percentage of total run time that the unit/volume/DDNAME combination used. There are two measures of run time:

- SOLO shows activity with neither concurrent CPU execution nor any other concurrent I/O activity.
- TOTAL shows activity with or without concurrent CPU execution or other I/O activity.

RUN TIME HISTOGRAM

Displays the intensity of activity within each unit/volume/DDNAME combination. The symbol "*" indicates solo usage of the device. The symbol "+" indicates the remaining nonsolo usage. Spikes, or lengthy lines in the histogram, draw your attention to DDNAMEs with a high proportion of activity.

MARGIN OF ERROR

The margin of error for run-time percentages. This percentage is the same value that appears in the RUN MARGIN OF ERROR PCT field on the Measurement Session Data report.

Most Intensively Executed Procedures Report

The Most Intensively Executed Procedures report (Figure 3-11) lists the ten most-used procedures in the programs executed in the measured job step. They are listed in descending order of CPU time used. CPU time for each procedure is given as a percentage of all CPU time. When CPU time is high, examine this report to select likely candidates for analysis in the Program Usage by Procedure reports (see Figure 3-15 on page 3-42 and Figure 3-16 on page 3-43). The report groups all system service routines by function under appropriately named pseudo-sections within the pseudo-module .SYSTEM. STROBE prints a brief description of a control section or a module if one is available.

Note: You can refer directly to this report when researching an opportunity to reduce CPU time. If the procedures listed on this report are using significant amounts of CPU time, you will not need to refer to the Program Section Usage Summary report and the Program Usage by Procedure report.

Figure 3-11. Most Intensively Executed Procedures Report

** MOST INTENSIVELY EXECUTED PROCEDURES **

MODULE NAME	SECTION NAME	LINE NUMBER	PROCEDURE/FUNCTION NAME	STARTING LOCATION	PROCEDURE LENGTH	CPU TIME PERCENT SOLO	CPU TIME PERCENT TOTAL	CUMULATIVE PERCENT SOLO	CUMULATIVE PERCENT TOTAL
STRBSAM1	SAMPLEW	70	WAY1-SUBSCRIPT	00096A	94	32.64	32.64	32.64	32.64
STRBSAM1	SAMPLEW	76	WAY2-INDEXED	0009C8	76	29.75	29.75	62.39	62.39
STRBSAM1	SAMPLEW	82	WAY3-ABS-SUB	000A14	96	22.31	22.31	84.70	84.70
STRBSAM1	SAMPLEW	89	WAY4-STRAIGHT	000A74	36	4.55	4.55	89.25	89.25
.SVC	SVC 019		OPEN			1.65	1.65	90.90	90.90
STRBSAM1	SAMPLEW	95	END-JOB	000A98	364	1.24	1.24	92.14	92.14
STRBSAM1	SAMPLEW	59	MAIN-3	00088E	220	1.24	1.24	93.38	93.38
STRBSAM1	SAMPLEW	54	MAIN-LINE	0007CE	192	1.24	1.24	94.62	94.62
.SVC	SVC 013		TERMINATION			1.24	1.24	95.86	95.86
.IOCS	IECOSCR1		PHYSICAL IOCS			1.24	1.24	97.10	97.10

MODULE NAME

The name of the load module in which STROBE detects resource usage. For module names that exceed eight characters, see the note in the SECTION NAME paragraph.

SECTION NAME

The name of the control section in the load module identified by MODULE NAME.

Note: For all module or section names that exceed eight characters, STROBE generates a *token*, which is an eight-byte identifier. The token comprises the first four characters of the module or section name followed by a hyphen (-) and then the last three characters of the name. Refer to the Token - Longname Cross Reference report, Figure 3-22 on page 3-52, to reconcile all tokens with their long names.

LINE NUMBER

The line in the program in which STROBE detects resource consumption. If you have not used STROBE to index the control section, this field is blank.

PROCEDURE/FUNCTION NAME

Procedures are identified by load module name, control section name, and starting location (in hexadecimal notation) within the control section. If a map data set or DDIO file has been used, the report also shows the source procedure name and line number. If you have not used STROBE to index the control section, the entries for line numbers and procedure names are blank. Such entries represent codeblocks of instructions equal in size to the report resolution. The report groups all system service routines by function under appropriately named pseudo-sections within the pseudo-module .SYSTEM.

STARTING LOCATION

The hexadecimal location within the named module and control section where a codeblock begins.

If you have not used STROBE to index the control sections, the entries represent codeblocks of instructions equal in size to the report resolution.

PROCEDURE LENGTH

The size of the procedure or codeblock in bytes, expressed in decimal notation.

CPU TIME PERCENT

The percentage of all CPU time used by programs executed in the measured job step. There are two measures of CPU time:

- SOLO shows activity without any concurrent I/O activity or concurrent task processing for the target program or subsystem.
- TOTAL shows activity with or without concurrent I/O activity.

CUMULATIVE PERCENT

The cumulative total of percentages in each column. There are two cumulative types of values:

- SOLO shows the cumulative total for each of the values in the SOLO column, for line entries down to and including itself.
- TOTAL shows the cumulative total for each of the values in the TOTAL column, for line entries down to and including itself.

Most Extensive Inactive Storage Areas Report

STROBE does not produce this report unless you request it. To obtain the report, supply the MEISA parameter when you produce a Performance Profile. To ensure that the information on the report is complete, you must specify"0" in the DETAIL BASELINE field on the STROBE - MODULE MAPPING panel when you submit the measurement request. For more information, see the *STROBE MVS User's Guide* or the *STROBE MVS User's Guide with Advanced Session Management*.

The Most Extensive Inactive Storage Areas report (Figure 3-12 on page 3-37) shows, by decreasing size, the ten largest areas of main storage that, although part of a program loaded by the measured job step, appear never to be executed. Portions of code that have outlived their usefulness and have been bypassed in the course of program evolution will very likely appear here. Large areas of inactive storage might cause inefficiencies in execution if the program is frequently loaded.

Because the measurement data is statistical, STROBE cannot ensure that an area is absolutely never used. You must analyze the program to determine whether an area of code is really never used or is rarely used. Code that is executed only at the end of month or year, for example, may appear on this report. Do not delete any code unless analyzing the program's flow reveals that it is never executed.

Each section in the Program Section Usage Summary report is treated in greater detail in the Program Usage by Procedure report (see Figure 3-15 on page 3-42).

Note: Information from Java programs does not appear on this report.

Figure 3-12. Most Extensive Inactive Storage Areas Report

** MOST EXTENSIVE INACTIVE STORAGE AREAS**

MODULE NAME	SECTION NAME	FROM LINE	WITHIN PROCEDURE	THRU LINE	WITHIN PROCEDURE	FROM LOCATION	THRU LOCATION	AREA LENGTH	CUMULATIVE LENGTH
DFHSAP	.PL/ILIB		IBMFPIR1		IBMFPIR1	000000	001607	5640	5640
DFHSAP	.PL/ILIB		IBMFERR1		IBMFERR1	000000	000C80	3201	8841
DFHSAP	.PL/ILIB		IBMFPGR1		IBMFPGR1	000000	000C22	3107	11948
STRBSAM1	.COBLIB		ILBOUST		ILBOUST	000000	00085F	2144	14092
DFHSAP	.PL/ILIB		IBMBLANU		IBMBLANU	000000	00074F	1872	15964
DFHSAP	.PL/ILIB		IBMBLANN		IBMBLANN	000000	00074F	1872	17836
DFHSAP	.PL/ILIB		IBMBLANA		IBMBLANA	000000	00074F	1872	19708
STRBSAM1	STREDYNA					000000	000711	1810	21518
STRBSAM1	.COBLIB		ILBOINS		ILBOINS	000000	0006E7	1768	23286
STRBSAM1	.COBLIB		ILBOPRM		ILBOPRM	000000	00066F	1648	24934

MODULE NAME

The name of the inactive load module. For module names that exceed eight characters, see the note in the SECTION NAME paragraph.

SECTION NAME

The name of the inactive control section. The report groups inactive compiler library routines included in the load module by library under an appropriately named pseudo-section. (For details about how to identify procedures, see "Indexed Performance Profiles" on page 2-26.)

Note: For all module or section names that exceed eight characters, STROBE generates a *token*, which is an eight-byte identifier. The token comprises the first four characters of the module or section name followed by a hyphen (-) and then the last three characters of the name. Refer to the Token - Longname Cross Reference report, Figure 3-22 on page 3-52, to reconcile all tokens with their long names.

FROM LINE

The beginning source text line number of the inactive area. If you have not indexed the control section, STROBE leaves this field blank.

WITHIN PROCEDURE

The name of the inactive procedure. If you have not used STROBE to index the control section, STROBE leaves this field blank.

THRU LINE

The ending source text line number of the inactive area. If you have not indexed the control section, STROBE leaves this field blank.

WITHIN PROCEDURE

The name of the inactive procedure. If you have not used STROBE to index the control section, STROBE leaves this field blank.

FROM LOCATION

The beginning storage location of the inactive area expressed in hexadecimal notation.

THRU LOCATION

The ending storage location of the inactive area expressed in hexadecimal notation.

AREA LENGTH

The length in bytes of the listed area, expressed in decimal notation.

CUMULATIVE LENGTH

The cumulative total for each of the values in the AREA LENGTH column, for line entries down to and including itself.

Program Section Usage Summary Report

The Program Section Usage Summary report (Figure 3-13 on page 3-39) shows the distribution of CPU time used by each active control section of each module in the target program or subsystem. When you create the Performance Profile and specify the ALLCSECT parameter, the report also displays inactive control sections. Examine this report to determine what programs or systems services are consuming the most CPU resources. You can then look up those programs in the Program Usage by Procedure report to determine which procedures cause the consumption.

Figure 3-13. Program Section Usage Summary Report

```
                                        ** PROGRAM SECTION USAGE SUMMARY **

 MODULE   SECTION 16M   SECT FUNCTION                 CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  6.30%
   NAME   NAME    <,>   SIZE                           SOLO     TOTAL        .00     23.50     47.00      70.50    94.00

.SYSTEM   .IOCS              DATA MANAGEMENT SERVICES   2.48      2.48        .*
.SYSTEM   .SVC               SUPERVISOR CONTROL         4.13      4.13        .*
                                                       -----     -----
.SYSTEM   TOTALS             SYSTEM SERVICES            6.61      6.61

STRBSAM1  .COBLIB      19670 COBOL LIBRARY SUBROUTINE    .41       .41        .
STRBSAM1  SAMPLEW       3328                           92.98     92.98        .****************************************
                                                       -----     -----
STRBSAM1  TOTALS   <   22998                           93.39     93.39
                                                       -----     -----
PROGRAM   STRBSAM1 TOTALS                             100.00    100.00
```

CPU time used by each control section is given as a percentage of all CPU time used in the measured job step; thus, the CPU time percentages used by all listed control sections total 100. The report orders entries alphabetically by module name and then section name.

The report groups all system service routines by function under appropriately named pseudo-sections within the pseudo-module .SYSTEM. It groups compiler library routines included in the target program or subsystem load modules under a suitably named pseudo-section.

MODULE NAME

The name of the module or pseudo-module for which STROBE is detailing CPU usage. For module names that exceed eight characters, see the note in the SECTION NAME paragraph.

SECTION NAME

The name of the control section or the pseudo-section within the module or pseudo-module identified in MODULE NAME for which STROBE is detailing CPU usage. A section name may not appear if the activity in the section is less than the percentage specified in the MODULE MAPPING BASELINE (default 2%) or if STROBE was unable to map the module. STROBE is unable to index a section that is not mapped.

Note: To ensure that the control section information is complete, you must compile all VisualAge products with the binder option EDIT=YES (the default).

For all module or section names that exceed eight characters, STROBE generates a *token*, which is an eight-byte identifier. The token comprises the first four characters of the module or section name followed by a hyphen (-) and then the last three characters of the name. Refer to the Token - Longname Cross Reference report, Figure 3-22 on page 3-52, to reconcile all tokens with their long names.

Note: STROBE reports "?LONG*nnn*" for the section names that exceed eight characters if initialization of the long names data space fails. The suffix "*nnn*" is a sequence number beginning with "001". The suffix is reset to "001" for each module. If initialization of the long names data space fails, STROBE also does not produce the Token - Longname Cross Reference report.

16M < , >

Indicates whether the RMODE of a module is 24 bit (<) or 31 bit (>). If the module has multiple control sections, this indication appears next to TOTALS in the SECTION NAME column. If the module is a Generalized Object File Format (GOFF) split-RMODE module an “S” appears in this column. If a GOFF module contains multiple control sections, the RMODE indicator appears for each control section.

SECT SIZE

The size of the control section in bytes, expressed in decimal notation. The report does not show section size for pseudo-sections, such as .IOCS and .SVC, that group related system services together and summarize those services.

FUNCTION

A short description of the function of the control section or pseudo-section or of the module or pseudo-module. Function descriptors appear for all pseudo-sections and pseudo-modules and, if your STROBE system programmer has supplied them, for other control sections and modules as well.

CPU TIME PERCENT

The percentage of all CPU time executing on behalf of the control section. There are two measures of CPU time:

- SOLO shows activity without any concurrent I/O activity for the target program or subsystem.
- TOTAL shows activity with or without concurrent I/O activity.

If a module contains more than one control section, the report also shows subtotals by control section. The Program Usage by Procedure report, Figure 3-15 on page 3-42, treats each control section in the Program Section Usage Summary report in greater detail.

CPU TIME HISTOGRAM

Displays the intensity of CPU usage within each control section. Solo CPU time is indicated by the symbol "*". The remaining CPU time is indicated by the symbol "+". Spikes, or lengthy lines in the histogram, highlight control sections with a high proportion of activity.

MARGIN OF ERROR

The margin of error for the CPU time percentages, also reported on the Measurement Session Data report. This margin of error, which appears in the header line, applies only to the number of samples in which STROBE found the CPU to be active.

Transaction Summary Report

The Transaction Summary report (Figure 3-14 on page 3-41) shows the distribution of CPU usage among transactions processed by the target subsystem. STROBE produces the report when it encounters transaction identifiers in the sample data set. The report lists transactions in ascending order by transaction identification code. This report enables you to identify transactions that use a great proportion of CPU resources. It also guides you to control sections shown in the Transaction Usage by Control Section report when you suspect that a transaction's use of resources is high for its frequency or value.

When you measure a CICS region and produce the CICS Performance Supplement, STROBE expands the Transaction Summary report and places it in the CICS Performance Supplement. For a description of the expanded report, see the STROBE *STROBE CICS Feature*.

When you measure an IMS Message Processing region (MPR) or CA-IDMS online region, STROBE expands the report, adding columns for transaction counts and average service time. For a description of the expanded report, see the STROBE *STROBE IMS Feature* or the STROBE *STROBE CA-IDMS Feature*.

Figure 3-14. Transaction Summary Report

```
                                        ** TRANSACTION SUMMARY **
 TRANSACTION    FUNCTION              TRANSACTION   AVERAGE     % CPU TIME    CPU TIME HISTOGRAM  MARGIN OF ERROR:  6.07%
    NAME                                 COUNT      SERVICE    SOLO   TOTAL   .00     25.00     50.00     75.00    100.00
                                                    TIME

  .CICS         CICS SYSTEM SERVICES                          31.58  31.58   .**********************
  ACCT          ACCOUNT STATUS INQUIRY                         2.63   2.63   .*
  PINV          PARTS INVENTORY INQUIRY                       55.26  55.26   .****************************************
  SCAL                                                        10.53  10.53   .*******
                                                              -----  -----
PROGRAM DFHSIP   TOTALS                                      100.00 100.00
```

Each transaction appearing in the Transaction Summary report is treated in greater detail in the "Transaction Usage by Control Section Report", Figure 3-17 on page 3-45.

TRANSACTION NAME

A transaction, pseudo-transaction, or function name supplied by a data collector program. If the name contains unprintable characters, STROBE shows it in hexadecimal notation.

FUNCTION

A short description of the function of the transaction or pseudo-transaction. Function descriptors appear for all STROBE pseudo-transactions and for other transactions with descriptions supplied by the STROBE system programmer.

TRANSACTION COUNT

This column appears for measurements of IMS Message Processing regions (MPRs) or CA-IDMS online regions. It displays the number of times a transaction was executed during the measurement session.

AVERAGE SERVICE TIME

This column appears for measurements of IMS Message Processing regions (MPRs) or CA-IDMS online regions. It displays the average service time, in seconds, for a transaction. Service time represents the time the transaction is being processed or awaiting processing within the region.

CPU TIME PERCENT

The percentage of CPU time spent executing each transaction. The percentage reflects CPU time used by programs executing for the measured job step. There are two measures of CPU time:

- SOLO shows activity without any concurrent I/O activity being performed under control of programs executed for the measured job step.
- TOTAL shows activity with or without any such concurrent I/O activity.

CPU TIME HISTOGRAM

Displays the intensity of CPU use within each transaction. Solo CPU time is indicated by the symbol "*". The remaining CPU time is indicated by the symbol "+". Spikes, or lengthy lines in the histogram, highlight transactions with a high proportion of activity. Examine those transactions that show large spikes for CPU usage to see if they can be made to function more efficiently.

MARGIN OF ERROR

The margin of error for CPU time percentages. This value also appears in the CPU MARGIN OF ERROR PCT field on the Measurement Session Data report. (See "CPU MARGIN OF ERROR PCT" on page 3-9.)

Program Usage by Procedure Report

The Program Usage by Procedure report (Figure 3-15) details the time the CPU spent executing code within each area of each control section of each module of the program or subsystem. The report is ordered alphabetically by module and, within module, by control section. Examine this report when the Program Section Usage Summary or the Most Intensively Executed Procedures reports show a concentration of CPU use. This report normally appears in two formats: a report formatted for system modules and a report formatted for user-written modules.

Figure 3-15. Program Usage by Procedure Report for System Modules

```
                                        ** PROGRAM USAGE BY PROCEDURE **

.SYSTEM     SYSTEM SERVICES                      .IOCS     DATA MANAGEMENT SERVICES

 MODULE   SECTION        FUNCTION            INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  1.53%
   NAME      NAME                             LENGTH   SOLO     TOTAL        .00       .50      1.00      1.50      2.00

 IGG019AI           QSAM SMPL PUT LOCATE F/U      176    .32       .32        .******
 IGG0193B           QSAM PUT                    26856    .05       .05        .*
 IGG0201Z           DATA MANAGEMENT SERVICES     9248   1.20      1.20        .************************
                                                        -----     -----
          .IOCS     TOTALS                             1.57      1.57

          .SYSTEM     SYSTEM SERVICES                     .SVC      SUPERVISOR CONTROL

 MODULE   SECTION        FUNCTION            INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  1.53%
   NAME      NAME                             LENGTH   SOLO     TOTAL        .00      9.50     19.00     28.50     38.00

 SVC 011            TIME                           37.86     37.86        .****************************************
 SVC 012            PROGRAM MANAGER/SYNCH            .10       .10        .
 SVC 019            OPEN                            5.00      5.00        .*****
 SVC 020            CLOSE                           8.39      8.42        .********
 SVC 079            TASK MANAGER/STATUS              .02       .02        .
 SVC 083            SMF                              .78       .78        .
 SVC 117            DEB VALIDITY CHECKING            .20       .20        .
 SVC 123            PURGEDQ                          .02       .02        .
 SVC 130            RACHECK                         4.88      4.88        .*****
                                                    -----     -----
          .SVC      TOTALS                         57.25     57.28
```

Reports for System Modules

This report shows CPU usage for all system service routines under the pseudo-module .SYSTEM.

A report for a system module begins with a header line that shows the pseudo-module (.SYSTEM) and the pseudo-section, with a description of its function.

Detail lines show:

- MODULE NAME, the true module name or an SVC identified by number.
- SECTION NAME, the control section name (if STROBE obtained one during sampling).
- FUNCTION, the function descriptor of the control section (if available) or the function descriptor of the module. Function descriptors do not appear for STROBE Features not installed at your site.

Reports for User-Written Modules

A report for a user-written module (Figure 3-16 on page 3-43) begins with a header line that contains the following:

- MODULE, the name of the module
- SECTION, the control section name (if STROBE obtained one during sampling)

- FUNCTION, the function descriptor of the control section, if the STROBE system programmer has supplied one

Note: For all module or section names that exceed eight characters, STROBE generates a *token*, which is an eight-byte identifier. The token comprises the first four characters of the module or section name followed by a hyphen (-) and then the last three characters of the name. Refer to the Token - Longname Cross Reference report, Figure 3-22 on page 3-52, to reconcile all tokens with their long names.

This report groups compiler library routines included in the target program or subsystem by library under a pseudo-section name that corresponds to its function.

Figure 3-16. Program Usage by Procedure Report for User-Written Modules

```
 MODULE - STRBSAM1
 SECTION - SAMPLEW
                                    SOURCE LANGUAGE - ANS COBOL VS                                VC24X25

  LINE      PROCEDURE        STARTING    INTERVAL  CPU TIME PERCENT         CPU TIME HISTOGRAM  MARGIN OF ERROR:  6.30%
 NUMBER       NAME          LOCATION      LENGTH   SOLO     TOTAL          .00      6.50     13.00     19.50     26.00

        INIT1 CODE           000000         1998    .00       .00        - .
    54  MAIN-LINE            0007CE           96    .83       .83          .*
    57     PERFORM           00082E           48    .41       .41          .
    58     PERFORM           00085E           48    .00       .00          .
    59  MAIN-3               00088E           90    .83       .83          .*
    65     WRITE             0008E8           56    .41       .41          .
    65     WRITE             000920           74    .00       .00        - .
    70  WAY1-SUBSCRIPT       00096A           54  14.88     14.88          .**********************
    72     ADD               0009A0           40  17.77     17.77          .***************************
    76  WAY2-INDEXED         0009C8           48  24.38     24.38          .*************************************
    79     IF                0009F8           28   5.37      5.37          .********
    82  WAY3-ABS-SUB         000A14           54  10.74     10.74          .****************
    86     ADD               000A4A           42  11.57     11.57          .*****************
    89  WAY4-STRAIGHT        000A74           36   4.55      4.55          .*******
    95  END-JOB              000A98          338   1.24      1.24          .*
    97     STOP              000BEA          270    .00       .00        - .
                                                  -----     -----
SECTION  SAMPLEW   TOTALS                        92.98     92.98
```

SOURCE LANGUAGE

The name of the language in which the section was programmed, such as ANSI COBOL VS. The value appears if you provided a map data set or DDIO file for the control section.

Control Code

The level number of the compiler and the generation of the STROBE Indexer used to produce the map data set. The code is given in the format VC*vcXgg*, where *vc* indicates the level number of the compiler and *gg* indicates the level of the STROBE Indexer used to produce the map data set. If you provided a map data set for the control section, this code appears at the far right of the subheading.

Procedure Identification for Indexed Control Sections

If a control section has been indexed, the report also shows:

- LINE NUMBER, the number assigned by the compiler to a line in the source program
- PROCEDURE NAME, a label derived from the source program that identifies the subsequent statement or statements

The rules for identifying procedures in this report depend on source language and are specified in “Indexed Performance Profiles” on page 2-26. This information guides you to the compiled source listing, so that you can analyze alternatives for ways to code procedures that invoke a high percentage of CPU resources.

STARTING LOCATION

The first storage location of the area, relative to the beginning of the control section, expressed in hexadecimal notation.

INTERVAL LENGTH

The length in bytes of the area covered by the entry line, expressed in decimal notation.

CPU TIME PERCENT

Each detail entry records the solo and total CPU time attributable to an area of code identified by line number and procedure name. CPU time for each area is given as a percentage of all CPU time used within the address space. There are two measures of CPU time:

- SOLO shows activity without any concurrent I/O activity for the target program or subsystem.
- TOTAL shows activity with or without concurrent I/O activity.

Totals for solo and total CPU time are presented after the detail entries for each control section. These totals appear in the "Program Section Usage Summary Report" (Figure 3-13 on page 3-39).

Note: Specifying the PUBP parameter when you create the Performance Profile combines values for control sections that use less than a given percentage of CPU time and can significantly shorten the length of the report that you need to review.

CPU TIME HISTOGRAM

Displays the intensity of CPU usage within neighboring areas of each procedure. Solo CPU time is indicated by the symbol "*". The remaining CPU time is indicated by the symbol "+". If the report combines some areas because they are inactive, the symbol "-" precedes the histogram line. Spikes, or lengthy lines in the histogram, highlight procedures with a high proportion of activity.

Note: Specifying the PUBP parameter when you create the Performance Profile suppresses the histogram for combined lines.

MARGIN OF ERROR

The margin of error for CPU time percentages. This value also appears in the CPU MARGIN OF ERROR PCT field on the Measurement Session Data report. (See "CPU MARGIN OF ERROR PCT" on page 3-9.)

Transaction Usage by Control Section Report

STROBE produces the Transaction Usage by Control Section report (Figure 3-17 on page 3-45) if it records transaction identifiers in the sample data set. The report lists transactions in ascending order by transaction identification code, module name, and control section name. If the transaction name contains unprintable characters, STROBE prints it in hexadecimal notation.

Figure 3-17. Transaction Usage by Control Section Report

```
TRANSACTION PINV             PARTS INVENTORY INQUIRY

 MODULE   SECTION                                    CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR: 15.90%
  NAME     NAME   COMPRESSED   FUNCTION               SOLO     TOTAL        .00     11.00     22.00     33.00     44.00

 .CICS    DFHSIP   DFHTRPX  TR DOM-TRC PUT-FAST PATH   2.63      2.63        .**
 .CICS    DFHSIP   KETIX    CICS COMPOSITE ROUTINES    2.63      2.63        .**
 .COBLIB  IGZCPAC  IGZCPRS  PROGRAM SETUP              2.63      2.63        .**
 DBVIORTN GETNEXTF                                    2.63      2.63        .**
 FINDPART SRCHLIST                                   42.11     42.11        .****************************************
 IGC253                     SUPERVISOR SERVICES       2.63      2.63        .**
                                                      -----     -----
TRANSACTION PINV      TOTALS                        55.26     55.26
```

When you measure a CICS region and produce the CICS Performance Supplement, the Transaction Usage by Control Section report appears, with a different format, in the CICS Performance Supplement. For a description of the report, see the STROBE *CICS Feature*.

Note: When you create the Performance Profile, you can supply the TUCS parameter to specify a threshold percentage of CPU time use, which can significantly shorten the length of the report that you need to review. With this parameter, the report omits transactions that use less than the amount you specify.

MODULE NAME

The name of the module or pseudo-module for which STROBE is detailing CPU use. For module names that exceed eight characters, see the note in the SECTION NAME paragraph.

SECTION NAME

The true name of the module appears in this column if the module is compressed. Otherwise, this column shows the control section name.

Note: For all module or section names that exceed eight characters, STROBE generates a *token*, which is an eight-byte identifier. The token comprises the first four characters of the module or section name followed by a hyphen (-) and then the last three characters of the name. Refer to the Token - Longname Cross Reference report, Figure 3-22 on page 3-52, to reconcile all tokens with their long names.

COMPRESSED

The name of the control section of the module that has been compressed within a pseudo-module.

FUNCTION

A short description of the function of the control section or pseudo-section or of the module or pseudo-module. Function descriptors appear for all pseudo-sections and pseudo-modules and, if your STROBE system programmer has supplied them, for other control sections and modules as well.

CPU TIME PERCENT

The percentage of all CPU time used on behalf of each control section of each module executed in the course of processing the transaction. There are two measures of CPU time:

- SOLO shows activity without any concurrent I/O activity being performed under control of programs executing within the address space.
- TOTAL shows activity with or without any concurrent I/O activity.

CPU TIME HISTOGRAM

The histogram shows the distribution of CPU usage within each control section. Solo CPU time is indicated by the symbol "*". The remaining CPU time is indicated by the

symbol "+". Spikes, or lengthy lines in the histogram, highlight control sections with a high proportion of activity.

MARGIN OF ERROR

The margin of error for CPU time percentages. This value also appears in the CPU MARGIN OF ERROR PCT field on the Measurement Session Data report. (See "CPU MARGIN OF ERROR PCT" on page 3-9.)

Coupling Facility Activity Report

The Coupling Facility Activity report provides a summary of coupling facility use for the entire system during the measurement session. The coupling facility activity shown on this report is independent of the STROBE measurement and is not directly related to activity in the measured program. The report is not based upon STROBE sampling data, and it reflects coupling facility use for the entire system. The report may identify a system-wide performance improvement opportunity that warrants further analysis. All statistics on this report apply to system-wide coupling facility activity.

Figure 3-18. Coupline Facility Activity Report

```
                                ** COUPLING FACILITY ACTIVITY **


---------------------------------ACROSS DURATION OF MEASUREMENT-----------------------------------------------------
                                                AVERAGE        AVERAGE                      TOTAL
COUPLING                                      SUCCESSFUL       FAILED       SUBCHANNEL     SUBCHANNEL        CF
FACILITY     TOTAL     SUCCESSFUL     FAILED     REQ TIME      REQ TIME      CONTENTION     CONTENTION    PROCESSOR
     NAME          REQ          REQ          REQ       MICRO SEC     MICRO SEC          COUNT        TIME  SEC   UTILIZATION
COUPL1       12,345       32,512     8123,456        2,558         1,776         12,112       2,125.897       82.5%
COUPL512    987,456          112        5,899          122           551             16            .854        1.9%
COUPLZZ9   9876,123     5648,258     5566,222     9911,254      1245,541       6413,874      88,221.366      100.0%
```

COUPLING FACILITY NAME

The name of the coupling facilities that were active during the STROBE measurement session.

TOTAL REQ

The total number of coupling facility requests made during the measurement session.

Note: The number in the TOTAL REQ field may not be the sum of the numbers in the SUCCESSFUL REQ and FAILED REQ fields. Because the TOTAL REQ field includes structures, XCF, XES, and failed requests, while the SUCCESSFUL REQ field includes only structures, there may be a discrepancy between the two fields.

SUCCESSFUL REQ

The number of successful coupling facility requests made to the coupling facility.

FAILED REQ

The number of coupling facility requests that failed for the coupling facility.

AVERAGE SUCCESSFUL REQ TIME

The average elapsed time, in microseconds, for a successful request to the coupling facility.

AVERAGE FAILED REQ TIME

The average elapsed time, in microseconds, for a failed request to the coupling facility.

SUBCHANNEL CONTENTION COUNT

The number of times a coupling facility request had to wait to be serviced because all subchannels for the coupling facility were busy.

TOTAL SUBCHANNEL CONTENTION TIME

The average elapsed time, in seconds, that a coupling facility request was in a wait state because all subchannels for the coupling facility were busy.

CF PROCESSOR UTILIZATION

Shows the percent of time that the processor for the coupling facility was active during the measurement session.

DASD Usage by Cylinder Report

The DASD Usage by Cylinder report (Figure 3-19 on page 3-48) presents a detailed analysis of the time that the channel program spent accessing each cylinder of each direct access device while the target program executed. This report can pinpoint inefficiencies caused by the way data is arranged on direct access devices.

Each section details activity on a separate direct access device, in numerical order by unit number. For each section, a header line identifies the device.

Within each section, detail lines classify activity on the device by cylinder, in ascending order by cylinder number.

The report identifies cylinders within data sets, and volume IDs, and identifies the DDNAME. In this way, a group of consecutive detail lines presents activity data for a given volume, while the activity for a single data set allocated to more than one volume can appear dispersed among the detail lines for those volumes.

Figure 3-19. DASD Usage by Cylinder Report

```
                                        ** DASD USAGE BY CYLINDER **

DEVICE ADDRESS  -  48A   TYPE - 3380

  VOLUME          DDNAME            CYLINDER          RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:  7.85%
    ID                                NUMBER            SOLO     TOTAL        .00      1.00      2.00      3.00      4.00

  WPA004         SYSUT1                60               .00      3.85        .+++++++++++++++++++++++++++++++++++++++
                                                        -----     -----
DEVICE ADDRESS  -  48A TOTALS                            .00      3.85

DEVICE ADDRESS  -  49C   TYPE - 3380

  VOLUME          DDNAME            CYLINDER          RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:  7.85%
    ID                                NUMBER            SOLO     TOTAL        .00      4.00      8.00     12.00     16.00

  WPA002         .FILEMGT               0       VTOC  14.10     14.10        .***********************************

                 SYSUT1              1253               .64       .64        .*
                 SYSUT1              1255               .64      4.49        .*++++++++++
                                                       -----     -----
DEVICE ADDRESS  -  49C TOTALS                          15.38     19.23

DEVICE ADDRESS  -  49D   TYPE - 3380

  VOLUME          DDNAME            CYLINDER          RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:  7.85%
    ID                                NUMBER            SOLO     TOTAL        .00      2.50      5.00      7.50     10.00

  WPA003         SYSUT1              2652              4.49      8.97        .*****************++++++++++++++++++
                                                       -----     -----
DEVICE ADDRESS  -  49D TOTALS                           4.49      8.97

DEVICE ADDRESS  -  747   TYPE - 3380

  VOLUME          DDNAME            CYLINDER          RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:  7.85%
    ID                                NUMBER            SOLO     TOTAL        .00      1.00      2.00      3.00      4.00

  S1RES1         .FILEMGT               0       VTOC   1.92      2.56        .********************++++++

                 SYSUT1              1521              1.92      1.92        .*******************
                                                       -----     -----
DEVICE ADDRESS  -  747 TOTALS                           3.85      4.49
```

Totals for each device follow the last entry of that device. The I/O Facility Utilization Summary report (Figure 3-10 on page 3-34) gives the aggregate solo and total times for each unit-volume-DDNAME.

Note: With the DASD parameter, you can combine cylinders displaying less than a certain percentage of DASD usage, which can shorten the report if you supply a value greater than the default of 2%. When cylinder lines are combined, the histogram is suppressed. In addition, with the DASDGAP parameter you can specify the number of inactive contiguous cylinders that STROBE will pass over while continuing to accrue CPU usage. The larger the value in this field, the more concise the report.

DEVICE ADDRESS

The address of the device.

TYPE

The device type.

VOLUME ID

The identifying number of the volume.

DDNAME

The data definition name.

CYLINDER NUMBER

The cylinder on which activity is detected. For cylinders that are inactive, this may include a range of cylinders, depending upon the numbers you specified in the DASD and DASDGAP parameters.

If the cylinder range includes a Volume Table of Contents (VTOC) for the device, STROBE displays VTOC in the column to the right of CYLINDER NUMBER Information.

Cylinders that have activity for open and close operations report under the pseudo-activity identifier .FILEMGT in the DD column.

RUN TIME PERCENT

Displays, for each line entry in the report, the percentage of total run time that the cylinder has been accessed. There are two measures of run time:

- SOLO shows activity with neither concurrent CPU execution nor any other concurrent I/O activity in the target program or subsystem
- TOTAL shows activity with or without concurrent CPU execution or other I/O activity.

Totals for each device follow the last entry for the device. The "I/O Facility Utilization Summary Report" (Figure 3-10 on page 3-34) gives the aggregate solo and total times for each unit, volume, and DDNAME.

RUN TIME HISTOGRAM

Shows the intensity of access to localized areas of the direct access device. Solo access time is indicated by the symbol "*". The remaining nonsolo access time is indicated by the symbol "+". Spikes, or lengthy lines in the histogram, highlight direct access devices that experience a high proportion of activity. Specifying the DASD parameter when you create the Performance Profile, suppresses the histogram for combined lines.

MARGIN OF ERROR

The margin of error for the run-time percentages. This value also appears in the RUN MARGIN OF ERROR PCT field on the Measurement Session Data report. (See "RUN MARGIN OF ERROR PCT" on page 3-9.)

Attribution Reports

Attribution reports identify sites of invocation of selected service routines. STROBE produces attribution reports for both CPU wait time (Figure 3-20 on page 3-49) and CPU execution time (Figure 3-21 on page 3-50). Examine these reports when system services use a high proportion of CPU resources or cause wait. These reports identify source statements that cause the activity.

Figure 3-20. Attribution of CPU Wait Time Report

```
                                   ** ATTRIBUTION OF CPU WAIT TIME **

.SVC     SVC 019            OPEN
-------------------------WAS INVOKED BY---------------------   -----------------VIA----------------------- WAIT TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                  PAGE  TOTAL

         STRBSAM1 SAMPLEW    000798     53      OPEN              STRBSAM1 ILBOQIO  QSAM I/O                    .00  2.25
                                                                                                             ----- -----
                                                                                                               .00  2.25

.SVC     SVC 020            CLOSE
-------------------------WAS INVOKED BY---------------------   -----------------VIA----------------------- WAIT TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                  PAGE  TOTAL

         STRBSAM1 SAMPLEW    000B30     95   END-JOB             STRBSAM1 HIBLOCK                            .00   .13
         STRBSAM1 SAMPLEW    000B30     95   END-JOB             STRBSAM1 ILBOQIO  QSAM I/O                  .00  1.98
                                                                                                             ----- -----
                                                                                                               .00  2.11
```

Each section of the attribution report begins with a header line that identifies the invoked service routine. The header also shows the pseudo-module name, module name,

control section name (when available), and a function descriptor for either the control section or the module. Reports for pseudo-module names .COBLIB, .PL/ILIB, .SASLIB, .C370LIB and .SVC have the format described here. For other pseudo-module names, look in Appendix A, “Program Structure and STROBE Pseudo-Entities” to determine the subsystem that STROBE detected. The STROBE Feature manual for the measured subsystem explains how to interpret these specialized attribution reports.

Each report detail line identifies a location in a program from which the service routine was directly or indirectly invoked. The Attribution of CPU Execution Time report shows solo and total CPU time spent in the service routine while it was invoked from that location. The Attribution of CPU Wait Time report shows the page and program wait time spent in the service routine while it was invoked from that location.

Figure 3-21. Attributionof CPU Execution Time Report

```
                                   ** ATTRIBUTION OF CPU EXECUTION TIME **

.SVC     SVC 013            TERMINATION
-------------------------WAS INVOKED BY---------------------   -----------------VIA-----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                SOLO  TOTAL

         IGC003              0009AC                                                                        1.24  1.24
                                                                                                          ----- -----
                                                                                                           1.24  1.24

.SVC     SVC 019            OPEN
-------------------------WAS INVOKED BY---------------------   -----------------VIA-----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                SOLO  TOTAL

         STRBSAM1 SAMPLEW    000798     53      OPEN             STRBSAM1 ILBOQIO  QSAM I/O                1.65  1.65
                                                                                                          ----- -----
                                                                                                           1.65  1.65

.SVC     SVC 020            CLOSE
-------------------------WAS INVOKED BY---------------------   -----------------VIA-----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                SOLO  TOTAL

         STRBSAM1 SAMPLEW    000B30     95   END-JOB             STRBSAM1 ILBOQIO  QSAM I/O                 .41   .41
                                                                                                          ----- -----
                                                                                                            .41   .41


.SVC     SVC 083            SMF
-------------------------WAS INVOKED BY---------------------   -----------------VIA-----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                SOLO  TOTAL


         STRBSAM1 SAMPLEW    000B30     95   END-JOB             STRBSAM1 ILBOQIO  QSAM I/O                 .41   .41
                                                                                                          ----- -----
                                                                                                            .41   .41

.SVC     SVC 130            RACHECK
-------------------------WAS INVOKED BY---------------------   -----------------VIA-----------------------  CPU TIME %
XACTION  MODULE   SECTION    RETURN   LINE   PROCEDURE NAME       MODULE   SECTION  FUNCTION                SOLO  TOTAL


         STRBSAM1 SAMPLEW    000798     53      OPEN             STRBSAM1 ILBOQIO  QSAM I/O                 .41   .41
                                                                                                          ----- -----
                                                                                                            .41   .41
```

The location that invoked the service routine is listed under “was invoked by”. The location that invoked the service routine is generally within a user-written routine and is defined by transaction, module, control section, and return address. If the invoking module or control section was a system routine, the report shows a function descriptor of either the control section or the module. If the invoking control section was a user-written routine and index data was available for the invoking control section, the report gives the line number and procedure name as well.

If the control section has not directly called the service routine, then the routine first called by the invoking location is defined under “via” by module, control section, and a brief description of its function.

XACTION

The name of the transaction that invoked the service routine, if there was one.

MODULE

The name of the module that invoked the service routine. If the invoker is a system module, the report identifies it by pseudo-module name, module name, and control section name. For module names that exceed eight characters, see the note in the SECTION paragraph.

SECTION

The name of the control section that invoked the service routine.

Note: For all module or section names that exceed eight characters, STROBE generates a *token*, which is an eight-byte identifier. The token comprises the first four characters of the module or section name followed by a hyphen (-) and then the last three characters of the name. Refer to the Token - Longname Cross Reference report, Figure 3-22 on page 3-52, to reconcile all tokens with their long names.

RETURN

The location, in hexadecimal, where the application invoked the service routine.

LINE

The line number at which the application invoked the service routine, for an indexed Performance Profile.

PROCEDURE NAME

The name of the procedure in which the application invoked the service routine, for an indexed Performance Profile.

Note: For pseudo-sections, the RETURN, LINE, and PROCEDURE NAME columns have different meanings. The RETURN column gives the name of the control section within the pseudosection that invoked the service routine. The LINE and the PROCEDURE NAME columns together contain the function descriptor of the control section identified in the LINE column.

MODULE (under VIA)

The name of the module that invoked the service routine if it was not directly invoked.

SECTION (under VIA)

The name of the control section that invoked the service routine if it was not directly invoked. For wait only, if the invoked module controls file access activities, the report shows the data definition name of the data set currently serviced by the invoked module instead of the control section name.

FUNCTION

A brief description of the function of the control section or module that invoked the service routine if it was not directly invoked.

CPU TIME PERCENT

For the Attribution of CPU Execution Time reports, the percentage of all CPU execution time used by programs executed within the address space that was spent in the invoked system routine on behalf of the invoking routine. There are two measures of CPU time:

- SOLO shows activity without any concurrent I/O activity being performed under control of programs executing within the address space
- TOTAL shows activity with or without any concurrent I/O activity.

WAIT TIME PERCENT

For the Attribution of CPU Wait Time reports, the percentage of wait time spent in the invoked system routine on behalf of the invoking routine. The time value is the percentage of all wait time used by programs executed in the measured job step for both page wait and total wait.

Token - Longname Cross Reference Report

STROBE produces the Token - Longname Cross Reference report when it encounters one or more module or control section names that exceed eight characters. This report lists the shortened version of the name, referred to as the token, and the entire module or section name. There are two columns in the report: Token and Longname.

Figure 3-22. Token - Longname Cross Reference Report

```
                                    ** TOKEN - LONGNAME CROSS REFERENCE **
TOKEN     LONGNAME
/u/s-048  /u/sxm/oe/test_pgm_for_long_csect_eqt_048
test-x48  test_program_for_long_csect_name_equal_to_xxx48
```

TOKEN

Displays the shortened version of the long name. The token reflects the first four characters of the module or section name, a hyphen (-), and the last three characters of the module or section name. In cases where tokens are identical,STROBE creates a unique name by replacing the last character of the generated token with a number from 1 to 9. If the tokens are still identical, STROBE replaces the last two characters of the token with a number from 10 to 99. In a case where the tokens are still not unique, STROBE replaces the last three characters of the token with numbers from 100 to 999.

LONGNAME

Displays the entire long name.

z/OS Memory Objects Report

STROBE produces the z/OS Memory Objects Report when it encounters z/OS memory objects. It identifies memory objects created by an application that are present when measurement takes place. The first section of the report entitled “MEMORY OBJECTS - FIRST OBSERVATION lists the memory objects STROBE first detects during measurement, The first section of the report entitled “MEMORY OBJECTS - LAST OBSERVATION lists the memory objects STROBE detects during its last measurement observation. If STROBE detects a higher count of memory objects between the first and last observations, it lists data for these memory sections in the section entitled “MEMORY OBJECTS - HIGHEST OBSERVATION OF USABLE BYTES”.

Note: This report will never contain more than 40 memory objects.

Figure 3-23. z/OS Memory Objects Report

```
                                        ** z/OS MEMORY OBJECTS **

MEM LIMIT                   64 MB SOURCE - JCL

                                    MEMORY OBJECTS FIRST OBSERVATION

-START ADDRESS--    - END ADDRESS--      SPK   ----- USABLE MB -----    - HIGH WATER MARK MB -    ---- GUARD BYTES MB ----

0000000100200000     00000001007FFFFF    08                 6
0000000100C00000     0000000100FFFFFF    05                 4
                                           ------------------
                                                             10                              10                           6

 ** NET CHANGE AT LAST OBSERVATION       **                  54                              54                          34

                                  MEMORY OBJECTS - HIGHEST OBSERVATION OF USABLE BYTES

-START ADDRESS--    - END ADDRESS--      SPK   ----- USABLE MB -----    - HIGH WATER MARK MB -    ---- GUARD BYTES MB ----

0000000100200000     00000001007FFFFF    08                  6
0000000100C00000     0000000100FFFFFF    05                  4
0000000101200000     00000001017FFFFF    08                  6
0000000101C00000     0000000101FFFFFF    05                  4
0000000102200000     00000001027FFFFF    08                  6
0000000102C00000     0000000102FFFFFF    05                  4
0000000103200000     00000001037FFFFF    08                  6
0000000103C00000     0000000103FFFFFF    05                  4
0000000104200000     00000001047FFFFF    08                  6
0000000104C00000     0000000104FFFFFF    05                  4
0000000105200000     00000001057FFFFF    08                  6
0000000105C00000     0000000105FFFFFF    05                  4
0000000106400000     00000001067FFFFF    05                  4
                                           ------------------
                                                             64                              64                          40

 ** NET CHANGE FROM FIRST OBSERVATION **                     54                              54                          34

                                    MEMORY OBJECTS - LAST OBSERVATION

-START ADDRESS--    - END ADDRESS--      SPK   ----- USABLE MB -----    - HIGH WATER MARK MB -    ---- GUARD BYTES MB ----

0000000100200000     00000001007FFFFF    08                  6
0000000100C00000     0000000100FFFFFF    05                  4
0000000101200000     00000001017FFFFF    08                  6
0000000101C00000     0000000101FFFFFF    05                  4
0000000102200000     00000001027FFFFF    08                  6
0000000102C00000     0000000102FFFFFF    05                  4
0000000103200000     00000001037FFFFF    08                  6
0000000103C00000     0000000103FFFFFF    05                  4
0000000104200000     00000001047FFFFF    08                  6
0000000104C00000     0000000104FFFFFF    05                  4
0000000105200000     00000001057FFFFF    08                  6
0000000105C00000     0000000105FFFFFF    05                  4
0000000106400000     00000001067FFFFF    05                  4
                                           ------------------
                                                             64                              64                          40

 ** NET CHANGE FROM HIGHEST OBSERVATION **                   0                               0                           0
```

For each memory object, the report provides:

START ADDRESS

The starting address of the memory object.

END ADDRESS

The ending address of the memory object.

SPK

The storage protection key that was specified when the memory object was created.

USABLE MB

The amount of usable memory available for the object. This amount is the total minus the guard bytes specified for the object.

HIGH WATER MARK MB

The largest amount of 64-bit storage that has been allocated in the measured address space from address space creation. This value may not reflect observed values for the address space during the measurement. This value is not reported for batch jobs because it may not accurately reflect the actual 64-bit storage use of the current batch job.

GUARD BYTES MB

The total of guard bytes that is part of all observed memory objects.

Reports Produced by the STROBE Features

Your Performance Profile includes additional reports and feature-specific data if your installation includes any of the following STROBE Features :

- ADABAS/NATURAL
- CA-IDMS
- CICS
- COOL:Gen (formerly Composer/IEF)
- CSP
- DB2
- IMS
- Java
- MQSeries
- UNIX System Services

The corresponding STROBE Feature manual details how to interpret these Performance Profile reports.

With the STROBE Advanced Session Management Feature, you can print or export the measurement session history reports that are created and displayed by STROBE/ISPF. Examples of these reports are shown and described in Appendix A of th*e STROBE MVS User's Guide with Advanced Session Management*.

Subsystem Attribution Reports

For each STROBE subsystem Feature, STROBE produces attribution reports specific to the subsystem it supports. For a complete description of the reports, see the related STROBE feature manual.

STROBE CICS Feature Reports

STROBE produces two types of CICS-related reports: the CICS Performance Supplement and the CICS Transaction Profile.

CICS Performance Supplement

The CICS Performance Supplement provides CICS-specific reports showing time distribution of transaction activity, transaction activity, network activity, I/O activity, configuration parameters, and system-wide statistics.

Time Distribution of Transaction Activity Level Report

The Time Distribution of Transaction Activity Level report shows, for the duration of the measurement session, the interactive transaction load placed on the CICS region and the level of performance of CICS under the transaction load. This report comprises four subreports:

- **Time Distribution of Transaction Arrivals** shows how the arrival rate of interactive transactions varied during the session.
- **Time Distribution of Transaction Service Time** shows the amount of time, in seconds per input message, that CICS spent processing the transactions whose arrival distribution is shown in the Time Distribution of Transaction Arrivals subreport.
- **Time Distribution of Data Base Operations** shows the time variation of input and output operations performed by CICS on the database while processing the transactions.
- **Time Distribution of Transaction Inputs, Processing, and Outputs** shows, for each 1% of the measurement session, the major statistics of input operations, processing, and output operations.

Transaction Activity Report

The Transaction Activity report shows the details of processing for individual transactions and transaction processing programs. It contains the following subreports:

- **Transaction Summary** shows, for every transaction that was invoked during the measurement session, its type and class, invocation count, CPU time, service time statistics, and initial processing program.
- **Transaction Exceptions** shows, when an exceptional condition has occurred for a transaction during the measurement session, the values of exception counters for the transactions.
- **Processing Program Statistics** shows the CICS activity of programs named in the CICS Program Processing Table.
- **CPU Usage by Control Section for Transaction *name*** shows, for each transaction and each CICS pseudo-transaction, the CPU time spent in the control sections of the modules executed for the transaction.

Network Activity Report

The Network Activity report shows the data communication operations that CICS performed during the measurement session to collect transactions and input messages from terminals and to deliver output messages to terminals. It has two subreports:

- **Terminal Statistics** shows the terminal status and data communication activity for each terminal that was active during the measurement session.
- **Terminal Exceptions** shows the status and error counts for each terminal associated with an exceptional condition during the measurement session.

Input-Output Activity Report

The Input-Output Activity report shows the input and output activity for all CICS files that were active during the measurement session. It comprises four subreports:

- **Summary of Input-Output Operations** summarizes all input and output operations, except for communication activity, during the session.
- **Data Base Operations** shows for each type of data base (VSAM, BDAM, ISAM, and DL/I) the number and type of input and output operations for each file accessed during the session.
- **Number of Output Operations to Transient Data Queues** shows the number and type of output operations for every transient data queue to which CICS wrote a record during the measurement session.
- **Journal Statistics** shows the output operations for each journal file that was active during the measurement session.

Configuration Parameters and System-Wide Statistics Report

The Configuration Parameters and System-Wide Statistics report shows several counts that apply to the entire CICS region and the values of several CICS parameters that affect the performance of the region. It has seven subreports:

- **Task Statistics** shows the number and types of tasks that CICS created to service the region.
- **Classed Tasks Statistics** shows the number and types of classed tasks that CICS created.
- **Storage Statistics** shows storage control statistics for main storage acquisitions and releases, failed conditional storage requests, storage violations and the incidence of other storage related occurrences.
- **VTAM Statistics** shows performance statistics for VTAM operations performed for the CICS region.
- **Temporary Storage Statistics** shows performance statistics for operations on CICS temporary storage.
- **Transient Data Statistics—Buffer and String Statistics** shows statistics related to buffers, strings, control intervals, and input/output operations on transient data queues.
- **Miscellaneous Statistics** shows statistics related to storage dumps and dynamic transaction backout operations.

CICS Transaction Profile

The CICS Transaction Profile provides CICS-specific reports that detail CICS transaction and region level information.

Transaction Summary

The Transaction Summary report is the first of the CICS transaction and region level reports. It provides high-level information for all user, mirror, system, or STROBE generated pseudo-transactions. For user and mirror transactions it attributes mean service time to the following categories: dispatch delay (the transaction is ready to execute but waiting to be dispatched), suspend (the transaction is waiting for an external resource), and execution (the transaction is running). For user and mirror transactions it also indicates whether detailed transaction information has been provided. It provides execution counts for user transactions, mirror transactions, and most system transactions. It shows CPU consumption for all transactions.

Transaction Profile Report

The Transaction Profile report helps you to target, identify, and understand performance problems at the transaction level. It consists of three subreports:

- CICS API Service Time shows, for a transaction, the subsystem API commands within a module and control section. The report shows the command, the resource that the command is targeting, the DBRM statement number (where applicable), the offset from which the command was issued, and the mean service time, which is attributed to suspended and executing states. If you have the STROBE DB2, STROBE IMS, STROBE Java, or STROBE MQSeries Features installed, STROBE will attribute the activity back to the command in the program that issued the subsystem call.
- CICS Non API Service Time accounts for service time that is not directly related to API commands. For resource type and resource name this report shows the dispatch delay and suspend time. For modules and control sections, it shows execution time.
- CICS API Service Time Detail provides further detail for the information provided in the CICS API Service Time report, breaking down suspend and execution time into more discrete components.

CICS Region Level Reports

The CICS region level reports provide an overall picture of suspend and dispatch delay time across an entire CICS address space. There are two region level reports:

- Region Suspend by Class provides the percentage of region suspend time and the average transaction suspend time for the various categories of suspend reasons.
- Region Suspend by Resource Within Class provides for each of the categories of suspend reasons, the resource type, the resource name, a description of the resource type that was suspended, the percentage of the region suspend time, and the average suspend time for a transaction.

The CICS Transaction Profile and CICS Region Level reports are produced only if the STROBE CICS Feature with the CICS Transaction Profile option is installed at your site. For more information, see the *STROBE CICS Feature*.

STROBE DB2 Feature Reports

- **SQL CPU Usage Summary** shows the distribution of CPU activity by DBRM or Query.
- **CPU Usage by SQL Statement** shows the distribution of CPU activity among SQL statements within a DBRM or Query. It displays the DBRM name in the header line, the number of times each statement was executed, and the average elapsed time for the statement calls. The report also identifies sites of invocation by their precompiled statement number.
- **SQL Wait Summary** identifies the total wait time by DBRM Query.
- **Wait by SQL Statement** provides detailed information for each SQL statement within a DBRM or Query, including the DBRM name, the number of times each statement was executed, and the average elapsed time for the statement calls.

For more information, see the *STROBE DB2 Feature*.

STROBE IMS Feature Reports

- **DL/I CPU Summary s**hows a summary of all attributed DL/I CPU activity by transaction, module, section, and Program Specification Block (PSB) name.
- **DL/I Wait Usage Summary** shows a summary of all attributed DL/I wait by transaction, module, section, and PSB name.
- **CPU Usage by DL/I Request** shows the IMS CPU activity that the STROBE IMS Feature attributed to a DL/I call statement within a transaction, module, section, and PSB name. Each report includes a retrieved segment section and a function activity section.
- **The Wait by DL/I Request** shows the IMS wait that the STROBE IMS Feature attributed to a DL/I call statement within a transaction, module, section, and PSB name. Each report includes a retrieved segment section and a function activity section.

For more information, see the *STROBE IMS Feature*.

STROBE ADABAS/NATURAL Feature Reports

- **Callers' DB Usage Summary** shows the percentage of total CPU activity consumed by each caller of ADABAS services. It identifies callers by job name (for a batch job), TSO user ID (for a TSO region), or logical terminal name (for a CICS region).
- **Callers' DB Usage by Control Section** appears for each of the database users identified in the Callers' DB Usage Summary report. The report header line identifies the calling region, and each detail line shows solo and total CPU time spent on

behalf of the caller in either an ADABAS or IBM system module or in a user-written control section.

- **NATURAL Program CPU Usage Summary** shows the NATURAL programs responsible for CPU activity.
- **CPU Usage by NATURAL Program Statement** shows the distribution of the CPU activity among the statements that make up the NATURAL programs detected by the STROBE ADABAS/NATURAL Feature. The report identifies the use of "INCLUDEd"' or copied, code in a NATURAL program, indicating the statement number and the corresponding name of the INCLUDE member.

  The report also identifies DB2 activity that occurs on behalf of a NATURAL program, indicating the SQL statement numbers and the DBRM name associated with the NATURAL program. For NATURAL for DB2, STROBE produces the CPU Usage by SQL Statement, which together with the CPU Usage by NATURAL Program Statement provides in-depth information on SQL statement CPU use.

  It also provides information about the IMS database that is being called by a NATURAL program.

For more information, see the *STROBE ADABAS/NATURAL Feature*.

STROBE CA-IDMS Feature Reports

- **Dialog CPU Summary** presents a summary of CPU usage by individual CA-ADS/O dialog. The report shows the dialog name, the transaction count and the CPU time.
- **CPU Usage by ADS/O Statement** details activity within each dialog process and lists the percent total for the entire dialog. The report identifies the dialog name in the header line and sites of invocation by their line number. Because CA-ADS/O dialogs can use "INCLUDEd", or copied, code, the report adds a suffix, indicating the nesting level of inclusion, to any line number that did not come from the mainline code.

For more information, see the *STROBE CA-IDMS Feature*.

STROBE CSP Feature Reports

The following reports are customized to reflect CSP-specific information:

- Program Section Usage Summary
- Most Intensively Executed Procedures
- Program Usage by Procedure
- Attribution of CPU Wait Time
- Attribution of CPU Execution Time

For more information, see the *STROBE CSP Feature*.

STROBE COOL:Gen Feature Reports

The Attribution of CPU Wait Time report and the Attribution of CPU Execution Time report are customized for the STROBE COOL:Gen Feature (formerly the STROBE Composer/IEF Feature) to reflect IEF-specific information. STROBE produces the following four new reports that summarize IEF activity:

- **IEF Action Diagram CPU Summary** shows the distribution of CPU activity among the action diagrams that STROBE identified during the measurement session.
- **CPU Usage by IEF Statement** shows the distribution of CPU activity among IEF statements within a specific action diagram.

- **IEF Action Diagram Wait Summary** identifies the total wait time by IEF action diagram.
- **Wait by IEF Statement** provides detailed information for each IEF statement within an action diagram.

For more information, see the *STROBE COOL:Gen Feature*.

STROBE Java Feature Reports

STROBE produces one set of reports for applications written using High Performance Java and a different set of reports for applications that run under the Java Virtual Machine (JVM).

HPJ Reports

STROBE produces the following Java reports for HPJ:

- **Java Class Summary** shows CPU activity and class size for classes and their respective modules present during the measurement.
- **CPU Usage by Java Method** shows CPU activity for Java methods within the classes and modules present during the measurement.
- **Attribution of CPU Execution** shows the Java method within the Java class that invoked a system service.

JVM Reports

STROBE products the following Java reports for JVM:

- **Java Targeting Information** shows this search argument(s) specified by the user to identify what Java application package/class(es) and methods should be measured.
- **Java Environment** shows the CLASSPATH information for each JVM profile in the measured address space.
- **Java CPU Usage by Called Method Summary** and **Java Wait Time by Called Method Summary** present Java package/class method execution and wait information at the method level
- **Java CPU Usage by Called Method** and **Java Wait Time by Called Method** show execution and attribution activity detected for Java methods and Java-invoked SQL statements.
- **Java CPU Usage by Execution Metho**d and **Java Wait Time by Called Method** capture and duplicate some of the CPU usage show in the Program Section Usage Summary and Program Usage by Procedure reports.

For more information see the *STROBE Java Feature*.

STROBE MQSeries Feature Reports

STROBE produces reports that detail MQSeries-related CPU usage and wait time incurred by a batch program. If your installation has the STROBE CICS Feature installed, the STROBE MQSeries Feature also reports on CICS- related service times by MQSeries Queue. Similarly, if your installation has the STROBE IMS Feature installed, STROBE can report IMS transaction-level activity within MQSeries modules.

- **MQSeries - CPU Usage Summary** shows CPU activity in modules and sections that used MQSeries services.
- **CPU Usage by Module by MQSeries Call** for each module shows individual service calls by storage location and for each storage location, the distribution of calls

among queues. If you have the STROBE IMS Feature and you specify the MQTRAN parameter, the report shows this information for each IMS transaction.

- **MQSeries - Total CPU Activity by Queue** shows a total of CPU activity by MQSeries Queue. This information is provided for all modules in the measured program. The report identifies the percent of CPU attributed to the queue that STROBE observed during sampling. It also provides data on options associated with the calls.
- **MQSeries Messages - CPU Activity by Queue by Module** provides data on the various modules that use a given queue and the characteristics of messages associated with that activity.
- **MQSeries - Wait Summary** shows a summary of the wait time experienced by application modules due to making MQSeries service calls.
- **Wait by Module by MQSeries Call** for modules with intensive MQSeries activity, shows the individual service calls by storage location. For each storage location, it shows the distribution of calls across queues. If you have the STROBE IMS Feature and you specify the MQTRAN parameter, the report shows this information for each IMS transaction.
- **MQSeries - Total Wait by Queue** shows a total of wait by MQSeries Queue. This information is provided for all modules in the measured program. The report identifies the percent of run time waiting number for MQSeries calls issued for the queue that STROBE observed during sampling. It also provides data on options associated with the calls.
- **MQSeries Messages - Wait by Queue by Module** shows total wait by queue and reports on the various modules that use a given queue and the characteristics of messages associated with that activity.
- **MQSeries - Service Time by Queue** provides CICS transaction wait, execution, and dispatch delay times by MQSeries Queue. This report appears only if you have the STROBE CICS Feature, are measuring a CICS MQSeries environment, and have selected to see Region Level CICS reports.
- **MQSeries - Summary of CPU Usage by Transaction** shows for each IMS transaction within an MQSeries module, solo and total percentages of CPU usage.
- **MQSeries - Summary of Wait Time by Transaction** summarizes for each IMS transaction within an MQSeries module, solo and total percentages of wait time.

For more information see the *STROBE MQSeries Feature*.

STROBE UNIX System Services Information

The STROBE UNIX System Services Feature provides information on applications and processes executing in IBM's OS/390 UNIX System Services address spaces. The STROBE Performance Profile is enhanced to show:

- CPU activity and wait time for modules running in OS/390 UNIX System Services address spaces
- the use of the Hierarchical File System (HFS) and zFS File System
- function descriptors for UNIX System Services modules

For more information, see the *STROBE UNIX System Services Feature*.

Appendix A.
Program Structure and STROBE Pseudo-Entities

STROBE defines a program or subsystem as all instructions executed during task execution for an address space. It is a complex structure that includes:

- instructions generated directly from user-coded source program statements during assembly or compilation
- instructions that make up IBM-supplied library routines included in the user's program load modules during linkage editing
- instructions in the IBM-supplied operating system routines that perform required supervisory and support functions

Figure A-1. Schematic Organization of a Program

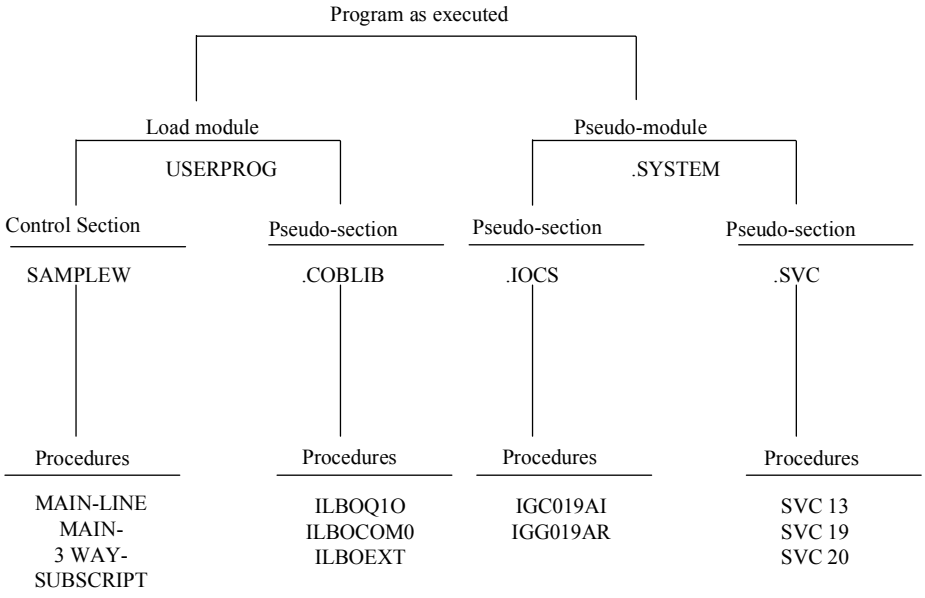


Figure A-1 contains a schematic representation of the structure of a simple program.

A complex batch processing program may be made up of many load modules invoked by one another during program execution. In the case of an online subsystem, the structure of each transaction processing program is similar to that shown in Figure A-1.

STROBE uses the following simplifications to make reports more concise and useful:

- Within a user program, STROBE attributes execution activity to a sequence of instructions that implement a source program procedure. When a STROBE Indexer is used, STROBE identifies the sequence of instructions by the programmer-assigned procedure name.
- STROBE treats library routines link-edited with a user program as procedures within library-specific pseudo-sections.

- STROBE treats IBM-supplied service modules as procedures within function-specific pseudo-sections. Function-specific pseudo-sections, in turn, are treated as components of the pseudo-module .SYSTEM.

Pseudo-Activity	Description
.FILEMGT	Catalog and index searches; open, close, and end-of-volume processing; and other data management activities, such as execution of certain supervisor calls and execution of access method modules (module names beginning with IGG) when the file being processed cannot be identified (as with some BDAM and ISAM modules).

Pseudo-Module	Description
.SYSTEM	Supervisor functions. The resource usage is attributed to the component pseudo-sections of this pseudo-module, as defined below.

Pseudo-Section	Description
.ADABAS	ADABAS database activity (module names beginning with ADA).
.ADASQL	ADABAS SQL Server statements (module names beginning with ESQ).
.AMBLIST	List service aid activity (module names beginning with AMB).
.AMS	Activity in Access Method Services modules (module names beginning with IDC).
.APPC	Activity in Advanced Program-to-Program Communication (APPC) scheduler modules (module names beginning with ASB and ATB).
.C370LIB	IBM C/370 modules (module names beginning with EDC and EDS).
.CDSA	Activity within the CICS dynamic storage area storage pool that is not within an identifiable module (for CICS V3.3 and higher).
.CICS	CICS modules (module names beginning with DFH).
.CLB3	Activity in C++ DLL routines.
.COBLIB	COBOL library modules (module names beginning with IGZ, IHD, or ILB).
.COMMON	Activity in the system common area, including the CSA, system queue area, and link pack areas.
.COMMONX	Activity in the extended system common areas above the 16-megabyte address line for MVS/XA and MVS/ESA, including the CSA, system queue area, and link pack areas.
.COMPRES	Cumulative mean service time for CICS resources and commands that is less than the specified baseline.
.COMSERV	Communications server for OS/390 activity (module names beginning with EZA, EZB, EZY, EZZ, and IKT).
.CSPLIB	CSP run-time library modules (modules with names beginning with ELA, FZE, or DZG).

Pseudo-Section	Description
.DB2	DB2 modules (module names beginning with DSN).
.DELAY	Mean service time when a CICS mirror or user transaction is ready to execute but the MVS task control block on which it needs to run is servicing another transaction.
.DMS	Display Management Service modules (module names beginning with DMX).
.DSA	Identifies activity within the CICS dynamic storage area storage pool that is not within an identifiable module (for CICS V1.7, V2.1, V3.1, and V3.2).
.DSM	Distributed Security Manager for MVS activity (module names beginning with EXR, EXS and FMH).
.ECDSA	Activity within the extended CICS dynamic storage area storage pool that is not within an identifiable module (for CICS V3.3 and higher).
.EDSA	Activity within the CICS extended user dynamic storage area storage pool that is not within an identifiable module (for CICS V3.1 and V3.2).
.EMULATE	Emulation modules (module names beginning with IIP, IIQ, IIR, or IIS).
.ERDSA	Activity within the CICS extended read-only dynamic storage area storage pool that is not within an identifiable module (for CICS V3.3 and higher).
.ESDSA	Activity within the CICS extended shared dynamic storage area storage pool that is not within an identifiable module (for CICS V4.1).
.EUDSA	Activity within the CICS extended user dynamic storage area storage pool that is not within an identifiable module (for CICS V3.3 and higher).
.FETCH	Identifies activity within program Fetch.
.FORTLIB	FORTRAN library modules (module names beginning with AFB, IHC, IHO, IFY, or IHN).
.GTF	Activity in Generalized Trace Facility (GTF) modules (module names beginning with AHL).
.HFS	Wait within Hierarchical File System (HFS) or Hierarchical File System (HFS) Adapter files.
.HPJ	Activity within High Performance Java modules.
.IAM	Modules supporting Innovation Access Method (IAM).
.IDMS	CA-IDMS modules (module names beginning with IDMS or RHDC).
.IEFLIB	IEF run-time library modules (module names beginning with TIR).
.IGW	Activity in OS/390 System Services file management routines.
.IMS	IMS modules (module names beginning with DFS, DBF, or DLZ).

Pseudo-Section	Description
.IOCS	Access method modules (module names beginning with IGG or IEC), such as GET and PUT modules.
.INVALID	Invalid characters found in module names.
.IPCS	Activity in Interactive Problem Control System modules (module names beginning with BLR and BLS).
.IRB	Asynchronous I/O error-recovery routines (routines executed under the control of an IRB).
.IRLM	Internal resource lock manager (module names beginning with DXR).
.ISG	Activity in Global Resource Serialization (GRS) modules.
.JAVA	Activity for the JVM (Java Virtual Machine) and HPJ (High Performance Java) runtimes.
.JES2	Job Entry Subsystem 2 activity (module names beginning with HAS, IAS, and IAZ).
.JES3	Job Entry Subsystem 3 activity (module name s beginning with IAT).
.LELIB	LE/370 library modules (modules beginning with CEE).
.LKD/LDR	Linking and loading modules (module names beginning with IEW).
.LOKWAIT	Indicates that the suspended task cannot acquire a lock on a requested resource.
.MEDIAMG	Media Manager modules (module names beginning with ICY).
.MQSRIES	Activity in MQSeries modules.
.NATNUC	Activity in the NATURAL nucleus (module names beginning with ACM, NAT, NCI, and NDB).
.NFSS	Activity in network file system server (NFSS) modules (module names beginning with GFSA).
.NONCICS	Mean service time that a CICS transaction is suspended by a resource manager interface (RMI) such as a subsystem interface. Mean service time is reported under this pseudo-section when the associated STROBE feature is not installed.
.NOWORK	The amount of wait attributed to the CICS dispatcher waiting for new work (transactions) to process.
.NUCLEUS	Activity in the MVS nucleus, generally branch-entered by the target program.
.NUCLEUX	Activity in the MVS supervisor above the 16-megabyte address line.
.PATHNME	Activity in system services or user modules with names that exceed eight characters but for which STROBE could not identify a module name.
.PL/ILIB	PL/I library modules (module names beginning with IBM, IHE, or IEL) and VisualAge Pl/I modules.

Pseudo-Section	Description
.PRIVATE	Activity in the private area that STROBE cannot otherwise identify, such as code moved to an area acquired by a GETMAIN.
.PRIVATX	Activity in the extended private area above the 16-megabyte address line that STROBE cannot otherwise identify, such as code moved to an area acquired by a GETMAIN.
.PSA	Activity in the Prefixed Storage Area.
.QMF	Query Management Facility modules (module names beginning with DSQ).
.RACF	Resource Access Control Facility (RACF) activity (module names beginning with ICH and IRR).
.RDSA	Activity within the CICS read only dynamic storage area storage pool that is not within an identifiable module (for CICS V4.1).
.REXX	Activity in REXX Language modules (module names beginning with IRX).
.RMF	Resource Measurement Facility activity (RMF) (module names beginning with AMS and ERB).
.SASCLIB	SAS/C library modules (module names beginning with LCS, L$C).
.SADMP	Stand-alone dump activity (module names beginning with AMD).
.SDSA	Activity within the CICS shared dynamic storage area storage pool that is not within an identifiable module (for CICSV4.1).
.SMP	System Maintenance Program activity (module names beginning with AMA and BCN).
.SMS	Activity in System Managed Storage (SMS) modules.
.SORT	Sort modules (module names beginning with ICE, IER, IGH, or SYN).
.STROBEC	Activity in the extended private area below the 16-megabyte address line associated with the STROBE capture facility routine.
.STROBEX	Activity in the extended private area above the 16-megabyte address line associated with the STROBE capture facility routine.
.SUPERVS	Supervisor Services activity (module names beginning with IGC).
.SVC	Modules invoked by supervisor calls, that is, routines executed under control of a Supervisor Request Block (SVRB), such as GETMAIN, FREEMAIN, OPEN, CLOSE.
.TCAM	Telecommunications Access Method modules (module names beginning with IED).
.TSO	Time Sharing Option (TSO) and Time Sharing Option/Extended (TSO/E) activity (module names beginning with ADF, CHS, END, ICQ, IGX0, IHASU, IKJ, INM, IRX, JBB, and SGIK).
.UDSA	Activity within the CICS user dynamic storage area storage pool that is not within an identifiable module (for CICS V3.3 and higher).

Pseudo-Section	Description
.UNNAMED	Activity in modules whose names contain unprintable characters.
.USERTASK	The amount of wait attributed to CICS-managed tasks waiting for an event to complete. Examples of events include data set I/O, program loads, and timer expiration.
.USS	Activity in OS/390 UNIX System Services support routines (module names beginning with BOP, BPX, FOM, and FSUM).
.VLF	Virtual Lookaside Facility activity (module names beginning with COF).
.VSAM	Virtual Storage Access Method modules (module names beginning with IDA). **Note:** To obtain VSAM attribution you must set the STROBE installation parameter LPALIB to YES. See Chapter 3 of the STROBE *System Programmer's Guide*.
.VTAM	Virtual Telecommunications Access Method modules (module names beginning with IST).
.WLM	Workload Manager (WLM) activity (module names beginning with IWM).
.XES	Activity in coupling facility service routines for cross-system data sharing in a sysplex environment (module names beginning with IXL).
.*xx*-TCB	Always associated with.DELAY, "*xx*" identifies the dispatcher mode task control block that has been requested.
.XMEMORX	Activity in the extended private area of another address space above the 16 megabyte address line that STROBE cannot otherwise identify; generally, code moved to an area acquired by a GETMAIN.
.XMEMORY	Activity in the private area of another address space that STROBE cannot otherwise identify; generally, code moved to an area acquired by a GETMAIN.

Note: Pseudo-transactions are discussed in the applicable STROBE feature manuals.

Appendix B.
Example Program and STROBE Performance Profile

This appendix contains the COBOL application program used as a case study in Chapter 2, “Interpreting the STROBE Performance Profile”, TAXRTN, the external subroutine that it calls, and the Performance Profile that resulted from the STROBE measurement.

Example COBOL Programs

This section contains the COBOL program and external subroutine used in Chapter 2, “Interpreting the STROBE Performance Profile”.

PAYROLL0

```
000001               IDENTIFICATION DIVISION.
000002
000003               PROGRAM-ID. PAYROLL0.
000004
000005               AUTHOR.   A. PROGRAMMER
000006
000007              ****************************************************************
*
000008              *
000009              *  S T R O B E     T E S T    P R O G R A M
000010              *
000011              ****************************************************************
*
000012              *                                    :
000013              *  MODULE......PAYROLL0 -            : THIS PROGRAM CALCULATES
000014              *                                    : THE EMPLOYEE PAYROLL.
000015              *                                    :
000016              *  DATE........1 JAN 1996            :
000017              *                                    :
000018              *                                    :
000019              *===================================================================
=
000020              * DESCRIPTION:
000021              * THIS PROGRAM COMPUTES AND PRINTS PAYCHECKS BY READING IN AN
000022              * EMPLOYEE'S WEEKLY TIME SHEET, FINDING A MATCHING EMPLOYEE FILE
000023              * RECORD, AND PERFORMING THE NECESSARY PAYROLL/TAX/ETC.
000024              * CALCULATIONS.
000025              *
000026              * FILES:
000027              * THE PROGRAM REQUIRES THE FOLLOWING FILES:
000028              * 1. EMPLOYEE FILE: ONE RECORD FOR EACH EMPLOYEE LISTING
000029              *    EMPLOYEE NUMBER, SOCIAL SECURITY NUMBER, NAME, ADDRESS,
000030              *    RATE OF PAY, CUMULATIVE SALARY HISTORY, ETC.
000031              * 2. TIME SHEET FILE: ONE RECORD PER EMPLOYEE SHOWING NUMBER OF
000032              *    HOURS WORKED.
000033              * 3. PAYCHECK OUTPUT FILE: ONE PAYCHECK WHICH LISTS DATE,
000034              *    EMPLOYEE NAME, DOLLAR AMOUNT, AND NET/GROSS SALARY
000035              *    TOTALS FOR THE PAYCHECK STUB.
000036              * 4. TAX FILE: VSAM FILE CONTAINING INCOME TAX RATES THAT IS
000037              *    USED AS A REFERENCE FILE BY AN EXTERNAL TAX SUBROUTINE.
000038              *
000039              *PROGRAM LOGIC FLOW:
000040              * -OPEN ALL FILES.
000041              * -MAIN LOOP:
000042              *   -FOR EACH TIME SHEET RECORD:
000043              *     -GET THE MATCHING EMPLOYEE RECORD FROM THE EMPLOYEE FILE,
000044              *        IF NO MATCHING RECORD THEN ABEND.
000045              *     -CALCULATE EMPLOYEE GROSS PAY, MULTIPLYING EMPLOYEE
000046              *        HOURS BY RATE OF PAY.
000047              *     -DETERMINE TAX RATE BY CALLING EXTERNAL PROGRAM TAXRTN.
000048              *          NOTE: TAXRTN SIMULATES AN IN-HOUSE FRONT-END TO AN
000049              *          TO AN OUTSIDE THIRD-PARTY TAX SERVICE. TO SIMPLIFY
000050              *          THE EXAMPLE, ONLY STATE TAXES ARE COMPUTED.
000051              *     -FINISH CALCULATIONS (TAX, NET PAY, YTD FIGURES, ETC.)
000052              *     -UPDATE THE EMPLOYEE FILE, OVERWRITING THE OLD DATA
```

```
000053              *        IF UNABLE TO UPDATE FILE THEN ABEND.
000054              *     -WRITE OUT THE PAYCHECK RECORD, IF ERROR THEN ABEND.
000055              * -CLOSE ALL FILES.
000056              *================================================================
=
000057              *
000058              *  INPUT FILES:  IN-TIMESHEET-FILE     VSAM KSDS FILE
000059              *                IO-EMPLOYEE-FILE      VSAM KSDS FILE
000060              *
000061              *  OUTPUT FILES: IO-EMPLOYEE-FILE      VSAM KSDS FILE
000062              *                OUT-PAYCHECK-FILE     FLAT FILE
000063              *
000064              *================================================================
=
000066               ENVIRONMENT DIVISION.
000067               CONFIGURATION SECTION.
000068               SOURCE-COMPUTER. IBM-370.
000069               OBJECT-COMPUTER. IBM-370.
000070               INPUT-OUTPUT SECTION.
000071               FILE-CONTROL.
000072
000073                   SELECT IN-TIMESHEET-FILE ASSIGN UT-AS-TIMSHEET
000074                          ORGANIZATION IS SEQUENTIAL
000075                          ACCESS       IS SEQUENTIAL
000076                          FILE STATUS  IS WS-IN-TIMESHEET-FILE-STAT.
000077
000078                   SELECT IO-EMPLOYEE-FILE ASSIGN UT-DA-EMPLOYEE
000079                          ORGANIZATION IS INDEXED
000080                          ACCESS       IS RANDOM
000081                          RECORD KEY   IS IO-EMPLOYEE-KEY1-ENUM
000082                          FILE STATUS  IS WS-IO-EMPLOYEE-FILE-STAT.
000083
000084                   SELECT OUT-PAYCHECK-FILE ASSIGN UT-S-PAYCHECK
000085                          FILE STATUS  IS WS-OUT-PAYCHECK-FILE-STAT.
000086
000087                   SELECT OUT-ERROR-FILE    ASSIGN UT-S-ERROR
000088                          FILE STATUS       IS WS-OUT-ERROR-FILE-STAT.
000089
000090               DATA DIVISION.
000091               FILE SECTION.
000092               FD  IN-TIMESHEET-FILE.
000093
000094               01  IN-TS-REC                     PIC  X(35).
000095
000096               FD  IO-EMPLOYEE-FILE.
000097
000098               01  IO-EMPLOYEE-REC.
000099                   05  IO-EMPLOYEE-KEY1-ENUM     PIC  X(04).
000100                   05  FILLER                    PIC  X(146).
000101
000102               FD  OUT-PAYCHECK-FILE
000103                   RECORDING MODE IS F
000104                   BLOCK CONTAINS 0 RECORDS
000105                   LABEL RECORDS ARE STANDARD.
000106
000107               01  OUT-PAYCHECK-REC              PIC  X(131).
000108
000109               FD  OUT-ERROR-FILE
000110                   RECORDING MODE IS F
000111                   BLOCK CONTAINS 0 RECORDS
000112                   LABEL RECORDS ARE STANDARD.
000113
000114               01  OUT-ERROR-REC                 PIC  X(80).
000115
000116
000117               WORKING-STORAGE SECTION.
000118
000119               77  WS-ABEND-CODE                 PIC  S9(4) COMP.
000120
000121               01  VARS.
000122
000123                   05  WS-GROSS                PIC S9(4)V9(2) COMP-3 VALUE +0.
000124                   05  WS-NET                  PIC S9(4)V9(2) COMP-3 VALUE +0.
000125                   05  WS-TAX                  PIC S9(3)V9(2) COMP-3 VALUE +0.
000126                   05  WS-YTD-GROSS            PIC S9(6)V9(2) COMP-3 VALUE +0.
000127                   05  WS-YTD-NET              PIC S9(6)V9(2) COMP-3 VALUE +0.
000128                   05  WS-YTD-TAX              PIC S9(5)V9(2) COMP-3 VALUE +0.
000129
000130
000131                   05  WS-ABEND-READ-EMP         PIC S9(4) VALUE +1001.
000132                   05  WS-ABEND-READ-TS          PIC S9(4) VALUE +1002.
000133                   05  WS-ABEND-READ-BAD-KEY     PIC S9(4) VALUE +1003.
000134                   05  WS-ABEND-REWRITE          PIC S9(4) VALUE +1004.
000135                   05  WS-ABEND-REWRITE-BAD-KEY  PIC S9(4) VALUE +1005.
000136                   05  WS-ABEND-WRITE            PIC S9(4) VALUE +1006.
```

```
000137                  05  WS-SALARY                PIC 9(9)V9(2).
000138                  05  WS-SALARY-DIGIT-ARRAY REDEFINES WS-SALARY
000139                                               PIC 9(1) OCCURS 11 TIMES.
000140                  05  WS-DIGIT-POSITION        PIC S9(2) VALUE 0 COMP-3.
000141                  05  WS-PREV-POSITION         PIC S9(2) VALUE 0 COMP-3.
000142                  05  WS-PREV-1-DIGIT          PIC 9(1).
000143                  05  WS-PREV-2-DIGIT          PIC 9(1).
000144                  05  WS-CURRENT-DIGIT         PIC 9(1).
000145                  05  WS-DTABLE-OFFSET         PIC S9(2) VALUE +0 COMP-3.
000146                  05  WS-PTABLE-OFFSET         PIC  9(2) VALUE  0.
000147                  05  WS-LG-CURR-CHAR          PIC S9(2) VALUE +0 COMP-3.
000148                  05  WS-LOOP                  PIC S9(2) VALUE +0 COMP-3.
000149                  05  WS-LOOPX                 PIC S9(2) VALUE +0 COMP-3.
000150                  05  WS-CHECK-NUM             PIC S9(8) VALUE +0 COMP-3.
000151                  05  WS-DATE                  PIC X(8) VALUE ' '.
000152                  05  WS-DATE-ARRAY            REDEFINES WS-DATE PIC X(1)
000153                                                 OCCURS 8 TIMES.
000154                  05  WS-ZERO                  PIC S9(1) VALUE +0
000155                                                 USAGE IS DISPLAY.
000156                  05  WS-SIGNIFICANT-DIGITS    PIC S9(2) VALUE +9 COMP-3.
000157                  05  WS-NUM-LEADING-ZEROS     PIC S9(2) VALUE +0 COMP-3.
000158
000159              01  WS-PARMS.
000160                  05  WS-STATE                 PIC  X(2).
000161                  05  WS-TAX-RATE              PIC  V9(2).
000162
000163              01  WS-TABLES.
000164
000165                  05  WS-TABLE1.
000166                      10  FILLER  PIC X(12) VALUE 'ZERO      05'.
000167                      10  FILLER  PIC X(12) VALUE 'ONE       04'.
000168                      10  FILLER  PIC X(12) VALUE 'TWO       04'.
000169                      10  FILLER  PIC X(12) VALUE 'THREE     06'.
000170                      10  FILLER  PIC X(12) VALUE 'FOUR      05'.
000171                      10  FILLER  PIC X(12) VALUE 'FIVE      05'.
000172                      10  FILLER  PIC X(12) VALUE 'SIX       04'.
000173                      10  FILLER  PIC X(12) VALUE 'SEVEN     06'.
000174                      10  FILLER  PIC X(12) VALUE 'EIGHT     06'.
000175                      10  FILLER  PIC X(12) VALUE 'NINE      05'.
000176
000177                  05  WS-ONES-TABLE REDEFINES WS-TABLE1.
000178                      20  WS-ONES-ROW OCCURS 10 TIMES.
000179                          25  WS-ONES-WORD             PIC   X(10).
000180                          25  WS-ONES-ARRAY REDEFINES WS-ONES-WORD
000181                                                PIC X(1) OCCURS 10 TIMES.
000182                          25  WS-ONES-CHARS-IN-WORD    PIC   9(2).
000183
000184                  05  WS-TABLE2.
000185                      10  FILLER  PIC X(12) VALUE 'TEN       04'.
000186                      10  FILLER  PIC X(12) VALUE 'ELEVEN    07'.
000187                      10  FILLER  PIC X(12) VALUE 'TWELVE    07'.
000188                      10  FILLER  PIC X(12) VALUE 'THIRTEEN  09'.
000189                      10  FILLER  PIC X(12) VALUE 'FOURTEEN  09'.
000190                      10  FILLER  PIC X(12) VALUE 'FIFTEEN   08'.
000191                      10  FILLER  PIC X(12) VALUE 'SIXTEEN   08'.
000192                      10  FILLER  PIC X(12) VALUE 'SEVENTEEN 10'.
000193                      10  FILLER  PIC X(12) VALUE 'EIGHTEEN  09'.
000194                      10  FILLER  PIC X(12) VALUE 'NINETEEN  09'.
000195
000196                  05  WS-TEENS-TABLE REDEFINES WS-TABLE2.
000197                      20  WS-TEENS-ROW OCCURS 10 TIMES.
000198                          25  WS-TEENS-WORD            PIC   X(10).
000199                          25  WS-TEENS-ARRAY REDEFINES WS-TEENS-WORD
000200                                                PIC X(1) OCCURS 10 TIMES.
000201                          25  WS-TEENS-CHARS-IN-WORD   PIC   9(2).
000202
000203                  05  WS-TABLE3.
000204                      10  FILLER  PIC X(12) VALUE '          00'.
000205                      10  FILLER  PIC X(12) VALUE '          00'.
000206                      10  FILLER  PIC X(12) VALUE 'TWENTY    07'.
000207                      10  FILLER  PIC X(12) VALUE 'THIRTY    07'.
000208                      10  FILLER  PIC X(12) VALUE 'FORTY     06'.
000209                      10  FILLER  PIC X(12) VALUE 'FIFTY     06'.
000210                      10  FILLER  PIC X(12) VALUE 'SIXTY     06'.
000211                      10  FILLER  PIC X(12) VALUE 'SEVENTY   08'.
000212                      10  FILLER  PIC X(12) VALUE 'EIGHTY    07'.
000213                      10  FILLER  PIC X(12) VALUE 'NINETY    07'.
000214
000215                  05  WS-TENS-TABLE REDEFINES WS-TABLE3.
000216                      20  WS-TENS-ROW OCCURS 10 TIMES.
000217                          25  WS-TENS-WORD             PIC   X(10).
000218                          25  WS-TENS-ARRAY REDEFINES WS-TENS-WORD
000219                                                PIC X(1) OCCURS 10 TIMES.
000220                          25  WS-TENS-CHARS-IN-WORD    PIC   9(2).
000221
```

```
000222                  05  WS-TABLE4.
000223                      10  FILLER  PIC X(12) VALUE '          00'.
000224                      10  FILLER  PIC X(12) VALUE 'HUNDRED   08'.
000225                      10  FILLER  PIC X(12) VALUE '          00'.
000226                      10  FILLER  PIC X(12) VALUE 'MILL      05'.
000227                      10  FILLER  PIC X(12) VALUE 'HUNDRED   08'.
000228                      10  FILLER  PIC X(12) VALUE '          00'.
000229                      10  FILLER  PIC X(12) VALUE 'THOUS     06'.
000230                      10  FILLER  PIC X(12) VALUE 'HUNDRED   08'.
000231                      10  FILLER  PIC X(12) VALUE '          00'.
000232                      10  FILLER  PIC X(12) VALUE '          00'.
000233
000234                  05  WS-PLACE-TABLE REDEFINES WS-TABLE4.
000235                      20  WS-PLACE-ROW OCCURS 10 TIMES.
000236                          25  WS-PLACE-WORD             PIC   X(10).
000237                          25  WS-PLACE-ARRAY REDEFINES WS-PLACE-WORD
000238                                             PIC X(1) OCCURS 10 TIMES.
000239                          25  WS-PLACE-CHARS-IN-WORD    PIC   9(2).
000240
000241              01  WS-FILE-LAYOUTS.
000242
000243                  05  WS-TS-REC.
000244                      10  WS-TS-ENUM            PIC  X(04).
000245                      10  WS-TS-EMP-HRS         PIC  9(2)V9(2).
000246                      10  WS-TS-PAY-PERIOD      PIC  X(8).
000247                      10  FILLER                PIC  X(19).
000248
000249                  05  WS-EMPLOYEE-REC.
000250                      10  WS-EMPLOYEE-KEY1-ENUM PIC  X(04).
000251                      10  WS-EMPLOYEE-SSN       PIC  X(09).
000252                      10  WS-EMPLOYEE-NAME      PIC  X(20).
000253                      10  WS-EMPLOYEE-STREET    PIC  X(20).
000254                      10  WS-EMPLOYEE-CITY      PIC  X(20).
000255                      10  WS-EMPLOYEE-STATE     PIC  X(02).
000256                      10  WS-EMPLOYEE-ZIP       PIC  X(09).
000257                      10  WS-EMPLOYEE-PAYRATE   PIC  9(3)V9(2).
000258                      10  WS-EMPLOYEE-YTD-GROSS PIC  9(6)V9(2).
000259                      10  WS-EMPLOYEE-YTD-NET   PIC  9(6)V9(2).
000260                      10  WS-EMPLOYEE-YTD-TAX   PIC  9(5)V9(2).
000261                      10  FILLER                PIC  X(38).
000262
000263                  05  WS-PAYCHECK-REC.
000264                      10  FILLER                PIC  X(1) VALUE ' '.
000265                      10  WS-PAYCHECK-OUTAREA   PIC  X(130).
000266
000267                  05  WS-BLANKS.
000268                      10  FILLER                PIC  X(1).
000269                      10  FILLER                PIC  X(130) VALUE ' '.
000270
000271                  05  WS-ERR1-REC.
000272                      10  FILLER  PIC X(16) VALUE 'PGM ABEND CODE= '.
000273                      10  WS-ERR1-ABEND PIC X(4).
000274                      10  FILLER  PIC X(10) VALUE ', BAD KEY='.
000275                      10  WS-ERR1-KEY    PIC 99.
000276                      10  FILLER  PIC X(48) VALUE ' '.
000277                  05  WS-ERR2-REC.
000278                      10  FILLER  PIC X(16) VALUE 'PGM ABEND CODE= '.
000279                      10  WS-ERR2-ABEND PIC X(4).
000280                      10  FILLER  PIC X(14) VALUE ', I/O STATUS= '.
000281                      10  WS-ERR2-STATUS PIC 99.
000282                      10  FILLER  PIC X(48) VALUE ' '.
000283
000284                  05  WS-LINEA-CHECK.
000285                      10  FILLER PIC X(1)   VALUE '0'.
000286                      10  FILLER PIC X(81)  VALUE 'CONNECTICUT PAYROLL SERVICE
000287              -    '                                                  #'.
000288                      10  WS-LINEA-CHECK-NUM PIC 9(8) VALUE 0.
000289                      10  FILLER PIC X(26)  VALUE                    '
000290              -    '  CUR PERIOD GROSS: '.
000291                      10  WS-LINEA-CURR-GROSS PIC $***,***,**9.99.
000292
000293                  05  WS-LINEB-CHECK.
000294                      10  FILLER PIC X(1)   VALUE ' '.
000295                      10 FILLER PIC X(115) VALUE '400 HARTFORD BLVD.
000296              -    '
000297              -    '        CUR PERIOD TAXES:'.
000298                      10  WS-LINEB-CURR-TAX PIC $***,***,**9.99.
000299
......                 Additional source statements to define output records
000410
000411
000412
000413              01  WS-SWITCHES.
000414
000415                  05  WS-DONE-SW                PIC  X(1)  VALUE 'N'.
```

```
000416                      88  WS-DONE                          VALUE 'Y'.
000417                      88  WS-NOT-DONE                      VALUE 'N'.
000418
000419                  05  WS-IN-TIMESHEET-FILE-STAT PIC X(2).
000420                      88  WS-IN-TIMESHEET-FILE-OK          VALUE '00'.
000421                      88  WS-IN-TIMESHEET-FILE-EOF         VALUE '10'.
000422
000423                  05  WS-IO-EMPLOYEE-FILE-STAT PIC   X(2).
000424                      88  WS-IO-EMPLOYEE-FILE-OK           VALUE '00'.
000425                      88  WS-IO-EMPLOYEE-FILE-BAD-KEY      VALUE '23'.
000426
000427                  05  WS-OUT-PAYCHECK-FILE-STAT PIC  X(2).
000428                      88  WS-OUT-PAYCHECK-FILE-OK          VALUE '00'.
000429                      88  WS-OUT-PAYCHECK-FILE-EOF         VALUE '10'.
000430
000431                  05  WS-OUT-ERROR-FILE-STAT    PIC  X(2).
000432                      88  WS-OUT-ERROR-FILE-OK             VALUE '00'.
000433                      88  WS-OUT-ERROR-FILE-EOF            VALUE '10'.
000434
000435
000436             *-------------------------------------------------------*
000437             *                                                       *
000438             *    TOP OF PROGRAM LOGIC                               *
000439             *                                                       *
000440             *-------------------------------------------------------*
000441              PROCEDURE DIVISION.
000442
000443                  PERFORM 0100-INITIALIZATION
000444                     THRU 0100-EXIT.
000445                  PERFORM 0200-MAIN-LOOP
000446                     THRU 0200-EXIT UNTIL WS-DONE.
000447                  PERFORM 0900-CLEANUP
000448                     THRU 0900-EXIT.
000449                  GOBACK.
000450
000451             *-------------------------------------------------------*
000452             *                                                       *
000453             *           OPEN FILES FOR INPUT, IO, OUTPUT.           *
000454             *                                                       *
000455             *-------------------------------------------------------*
000456              0100-INITIALIZATION.
000457
000458                  OPEN INPUT  IN-TIMESHEET-FILE
000459                       I-O    IO-EMPLOYEE-FILE
000460                       OUTPUT OUT-PAYCHECK-FILE
000461                       OUTPUT OUT-ERROR-FILE.
000462
000463                  IF ( (NOT WS-IN-TIMESHEET-FILE-OK)  OR
000464                       (NOT WS-IO-EMPLOYEE-FILE-OK )  OR
000465                       (NOT WS-OUT-PAYCHECK-FILE-OK)  OR
000466                       (NOT WS-OUT-ERROR-FILE-OK)         )
000467                     DISPLAY 'TIMESHEET = ' WS-IN-TIMESHEET-FILE-STAT
000468                             ', EMPLOYEE  = ' WS-IO-EMPLOYEE-FILE-STAT
000469                             ', PAYCHECK  = ' WS-OUT-PAYCHECK-FILE-STAT
000470                             ', ERROR     = ' WS-OUT-ERROR-FILE-STAT
000471                     MOVE 'Y' TO WS-DONE-SW
000472                     GO TO 0100-EXIT
000473                  END-IF.
000474
000475             *-   GET DATE FROM COBOL "DATE" FUNCTION ----------------*
000476             *-   DATE FORMAT IS YYMMDD, MOVE TO MM/DD/YY-------------*
000477                  ACCEPT WS-DATE FROM DATE.
000478                  MOVE WS-DATE-ARRAY(1) TO WS-LINEC-DATE-ARRAY(7).
000479                  MOVE WS-DATE-ARRAY(2) TO WS-LINEC-DATE-ARRAY(8).
000480                  MOVE WS-DATE-ARRAY(3) TO WS-LINEC-DATE-ARRAY(1).
000481                  MOVE WS-DATE-ARRAY(4) TO WS-LINEC-DATE-ARRAY(2).
000482                  MOVE WS-DATE-ARRAY(5) TO WS-LINEC-DATE-ARRAY(4).
000483                  MOVE WS-DATE-ARRAY(6) TO WS-LINEC-DATE-ARRAY(5).
000484
000485             *-   INITIALIZE FIELDS IN TEMPLATES FOR CHECK LINES ------*
000486                  MOVE 0            TO WS-LINEA-CURR-GROSS.
000487                  MOVE 0            TO WS-LINEB-CURR-TAX.
000488                  MOVE 0            TO WS-LINEC-CURR-NET.
000489                  MOVE 0            TO WS-LINED-YTD-GROSS.
000490                  MOVE 0            TO WS-LINEE-SUM.
000491                  MOVE 0            TO WS-LINEE-YTD-TAX.
000492                  MOVE 0            TO WS-LINEF-YTD-NET.
000493                  MOVE 0            TO WS-LINEH-CENTS1.
000494                  MOVE 0            TO WS-LINEH-CENTS2.
000495
000496              0100-EXIT.
000497                  EXIT.
000498             *-------------------------------------------------------*
000499             *                                                       *
000500             *                       MAIN  LOOP                      *
```

```
000501              *                                                       *
000502              *-------------------------------------------------------*
000503               0200-MAIN-LOOP.
000504
000505                    PERFORM 0300-READ-TIMESHEET-FILE
000506                       THRU 0300-EXIT.
000507
000508                    IF WS-DONE
000509                       GO TO 0200-EXIT.
000510
000511                    PERFORM 0400-READ-EMPLOYEE-FILE
000512                       THRU 0400-EXIT.
000513
000514                    IF WS-TS-EMP-HRS = 0 THEN
000515                       GO TO 0200-EXIT.
000516
000517                    PERFORM 0500-PREP-CHECK
000518                       THRU 0500-EXIT.
000519
000520                    PERFORM 0700-WRITE-PAYCHECK-FILE
000521                       THRU 0700-EXIT.
000522
000523                    PERFORM 0800-REWRITE-EMPLOYEE-FILE
000524                       THRU 0800-EXIT.
000525
000526               0200-EXIT.
000527                    EXIT.
000528              *-------------------------------------------------------*
000529              *                                                       *
000530              *                  READ TIMESHEET FILE                  *
000531              *                                                       *
000532              *-------------------------------------------------------*
000533               0300-READ-TIMESHEET-FILE.
000534
000535                    READ IN-TIMESHEET-FILE INTO WS-TS-REC
000536                        AT END MOVE 'Y' TO  WS-DONE-SW
000537                               GO TO 0300-EXIT.
000538
000539                    IF (NOT WS-IN-TIMESHEET-FILE-OK)
000540                      MOVE WS-ABEND-READ-TS TO WS-ABEND-CODE
000541                      MOVE WS-ABEND-READ-TS TO WS-ERR2-ABEND
000542                      MOVE WS-IN-TIMESHEET-FILE-STAT TO WS-ERR2-STATUS
000543                      WRITE OUT-ERROR-REC FROM WS-ERR2-REC
000544                      PERFORM 0900-CLEANUP THRU 0900-EXIT
000545                      CALL 'ILBOABNO' USING WS-ABEND-CODE
000546                    END-IF.
000547
000548               0300-EXIT.
000549                    EXIT.
000550              *-------------------------------------------------------*
000551              *                                                       *
000552              *                    READ EMPLOYEE FILE                 *
000553              *                                                       *
000554              *-------------------------------------------------------*
000555               0400-READ-EMPLOYEE-FILE.
000556
000557                    MOVE WS-TS-ENUM TO IO-EMPLOYEE-KEY1-ENUM.
000558
000559                    READ IO-EMPLOYEE-FILE INTO WS-EMPLOYEE-REC
000560                       KEY IS IO-EMPLOYEE-KEY1-ENUM
000561                         INVALID KEY
000562                            MOVE WS-ABEND-READ-BAD-KEY TO WS-ABEND-CODE
000563                            MOVE WS-ABEND-READ-BAD-KEY TO WS-ERR1-ABEND
000564                            MOVE IO-EMPLOYEE-KEY1-ENUM TO WS-ERR1-KEY
000565                            WRITE OUT-ERROR-REC FROM WS-ERR1-REC
000566                            PERFORM 0900-CLEANUP THRU 0900-EXIT
000567                            CALL 'ILBOABNO' USING WS-ABEND-CODE.
000568
000569                    IF (NOT WS-IO-EMPLOYEE-FILE-OK)
000570                      MOVE WS-ABEND-READ-EMP TO WS-ABEND-CODE
000571                      MOVE WS-ABEND-READ-EMP TO WS-ERR2-ABEND
000572                      MOVE WS-IO-EMPLOYEE-FILE-STAT TO WS-ERR2-STATUS
000573                      WRITE OUT-ERROR-REC FROM WS-ERR2-REC
000574                      PERFORM 0900-CLEANUP THRU 0900-EXIT
000575                      CALL 'ILBOABNO' USING WS-ABEND-CODE
000576                    END-IF.
000577
000578               0400-EXIT.
000579                    EXIT.
000580              *-------------------------------------------------------*
000581              *                                                       *
000582              *                PREPARE CHECK                          *
000583              *                                                       *
000584              *-------------------------------------------------------*
000585               0500-PREP-CHECK.
```

```
000586
000587                  MOVE 0 TO WS-GROSS.
000588                  MOVE 0 TO WS-NET.
000589                  MOVE 0 TO WS-TAX.
000590                  MOVE 0 TO WS-YTD-GROSS.
000591                  MOVE 0 TO WS-YTD-NET.
000592                  MOVE 0 TO WS-YTD-TAX.
000593
000594
000595                  COMPUTE WS-GROSS =
000596                          WS-TS-EMP-HRS * WS-EMPLOYEE-PAYRATE.
000597
000598                  MOVE WS-EMPLOYEE-STATE TO WS-STATE.
000599
000600                  CALL 'TAXRTNO' USING BY REFERENCE WS-PARMS.
000601
000602                  COMPUTE WS-TAX =
000603                          WS-GROSS * WS-TAX-RATE.
000604
000605                  COMPUTE WS-NET =
000606                          WS-GROSS - WS-TAX.
000607
000608                  ADD  WS-GROSS          TO  WS-EMPLOYEE-YTD-GROSS.
000609                  ADD  WS-NET            TO  WS-EMPLOYEE-YTD-NET.
000610                  ADD  WS-TAX            TO  WS-EMPLOYEE-YTD-TAX.
000611
000612                  ADD 1 TO WS-CHECK-NUM.
000613                  MOVE WS-NET TO WS-SALARY.
000614
000615                  MOVE WS-LINEA-CHECK           TO WS-LA-CHECK.
000616                  MOVE WS-CHECK-NUM             TO WS-LA-CHECK-NUM.
000617                  MOVE WS-GROSS                 TO WS-LA-CURR-GROSS.
000618                  MOVE WS-LINEB-CHECK           TO WS-LB-CHECK.
000619                  MOVE WS-TAX                   TO WS-LB-CURR-TAX.
000620                  MOVE WS-LINEC-CHECK           TO WS-LC-CHECK.
000621                  MOVE WS-NET                   TO WS-LC-CURR-NET.
000622                  MOVE WS-LINED-CHECK           TO WS-LD-CHECK.
000623                  MOVE WS-EMPLOYEE-YTD-GROSS    TO WS-LD-YTD-GROSS.
000624                  MOVE WS-LINEE-CHECK           TO WS-LE-CHECK.
000625                  MOVE WS-EMPLOYEE-NAME         TO WS-LE-NAME.
000626                  MOVE WS-SALARY                TO WS-LE-SUM.
000627                  MOVE WS-EMPLOYEE-YTD-TAX      TO WS-LE-YTD-TAX.
000628                  MOVE WS-LINEF-CHECK           TO WS-LF-CHECK.
000629                  MOVE WS-EMPLOYEE-YTD-NET      TO WS-LF-YTD-NET.
000630                  MOVE WS-LINEG-CHECK           TO WS-LG-CHECK.
000631                  MOVE WS-LINEH-CHECK           TO WS-LH-CHECK.
000632
000633                  PERFORM 0600-TRANSLATE-CHECK THRU 0600-EXIT.
000634
000635           0500-EXIT.
000636                  EXIT.
000637           *-------------------------------------------------------*
000638           *                                                       *
000639           *                  TRANSLATE CHECK                      *
000640           *                                                       *
000641           *  THIS ROUTINE IS USED FOR CHECK FORMATTING, IT        *
000642           *  TRANSLATES THE SIGNIFICANT DIGITS OF THE SUM FIELD OF  *
000643           *  A CHECK (EX. $513.95) INTO THE CORRECT TEXT STRING (EX.*
000644           *  FIVE HUNDRED THIRTEEN AND 95/100 DOLLARS). THIS ROUTINE*
000645           *  WILL NEVER BE CALLED IF SUM PAYABLE <= 0.            *
000646           *                                                       *
000647           *  THIS ROUTINE (AND TRANSLATION ROUTINES CALLED BY IT)   *
000648           *  DEPEND UPON A SIMPLIFYING ASSUMPTION THAT THE CHECK    *
000649           *  SUM HAS A MAXIMUM OF 11 TOTAL DIGITS, GIVING *NINE*    *
000650           *  SIGNIFICANT DIGITS ($999,999,999.99).  EACH DIGIT IN   *
000651           *  THE WS-SALARY FIELD IS ASSIGNED A DIGIT-POSITION:      *
000652           *                                                       *
000653           *     SALARY:           999,999,999.99                  *
000654           *     DIGIT POSITION:   123 456 789 11                  *
000655           *                                 01                    *
000656           *                                                       *
000657           *  SO, DIGIT-POSITIONS 3,6, AND 9 ARE IN THE 'HUNDREDS'   *
000658           *  PLACE, POSITIONS 2,5, AND 8 ARE IN THE 'TENS' PLACE,   *
000659           *  POSITIONS 10 AND 11 ARE THE CENTS FIELDS, ETC.         *
000660           *                                                       *
000661           *  TABLES ARE USED TO TRANSLATE DIGITS AND PLACE VALUE    *
000662           *  NAMES, ACCORDING TO THE CURRENT-DIGIT (DTABLE) OR      *
000663           *  DIGIT-POSITION (PTABLE).  A TABLE ROW CONTAINS A TEXT  *
000664           *  FIELD (THE STRING TO MOVE, PADDED WITH BLANKS) AND A   *
000665           *  LENGTH FIELD (# OF CHARS TO MOVE).  FOR EXAMPLE, IF THE*
000666           *  TENS DIGIT IS '2', THEN MOVE TO THE CHECK OUTPUT AREA  *
000667           *  7 CHARACTERS (STRING "TWENTY ").  VARIABLE             *
000668           *  WS-LG-CURR-CHAR TRACKS THE LOCATION IN THE CHECK       *
000669           *  OUTPUT AREA TO WHICH THE TRANSLATION OF THE CURRENT-   *
000670           *  DIGIT SHOULD BE COPIED.                              *
```

```
000671              *                                                       *
000672              *  LOGIC:                                               *
000673              *    WS-LG-CURR-CHAR = 0                                *
000674              *    OBTAIN # OF LEADING ZEROS IN WS-SALARY             *
000675              *    DETERMINE # OF SIGNIFICANT DIGITS IN WS-SALARY     *
000676              *    IF # OF SIGNIFICANT DIGITS >= 10, PRINT "ZERO"     *
000677              *     (HANDLES CASE WHERE $0.01 <= WS-SALARY <= $0.99)  *
000678              *    ELSE                                               *
000679              *     FOR EACH SIGNIFICANT DIGIT CALL 0610-TRANSLATE-DIGIT*
000680              *    END-IF.                                            *
000681              *    INITIALIZE CENTS POSITIONS IN OUTPUT AREA          *
000682              *                                                       *
000683              *-------------------------------------------------------*
000684               0600-TRANSLATE-CHECK.
000685
000686                    MOVE 0 TO WS-NUM-LEADING-ZEROS.
000687                    MOVE 0 TO WS-LG-CURR-CHAR.
000688
000689                    INSPECT WS-SALARY
000690                      TALLYING WS-NUM-LEADING-ZEROS FOR LEADING WS-ZERO.
000691                    COMPUTE WS-DIGIT-POSITION = WS-NUM-LEADING-ZEROS + 1.
000692                    IF WS-DIGIT-POSITION <= WS-SIGNIFICANT-DIGITS THEN
000693                       PERFORM 0610-TRANSLATE-DIGIT
000694                          THRU 0610-EXIT
000695                          UNTIL WS-DIGIT-POSITION > WS-SIGNIFICANT-DIGITS
000696                    ELSE
000697                       MOVE 1 TO WS-DTABLE-OFFSET
000698                       PERFORM 0620-TRANSLATE-ONES THRU 0620-EXIT
000699                    END-IF.
000700
000701                    MOVE WS-SALARY-DIGIT-ARRAY(10) TO WS-LH-CENTS1.
000702                    MOVE WS-SALARY-DIGIT-ARRAY(11) TO WS-LH-CENTS2.
000703
000704               0600-EXIT.
000705                    EXIT.
000706              *-------------------------------------------------------*
000707              *                                                       *
000708              *                TRANSLATE DIGIT                        *
000709              *                                                       *
000710              * THIS ROUTINE DEPENDS UPON THE WS-SALARY FIELD DEFINED  *
000711              * FOR A MAXIMUM OF 11 DIGITS (SEE PREP-CHECK PROCEDURE). *
000712              *-------------------------------------------------------*
000713               0610-TRANSLATE-DIGIT.
000714
000715                    MOVE WS-SALARY-DIGIT-ARRAY(WS-DIGIT-POSITION)
000716                      TO WS-CURRENT-DIGIT.
000717
000718                    COMPUTE WS-DTABLE-OFFSET  = WS-CURRENT-DIGIT + 1.
000719                    COMPUTE WS-PTABLE-OFFSET  = WS-DIGIT-POSITION + 1.
000720
000721                    IF WS-DIGIT-POSITION > 1 THEN
000722                        COMPUTE WS-PREV-POSITION = WS-DIGIT-POSITION - 1
000723                        MOVE WS-SALARY-DIGIT-ARRAY(WS-PREV-POSITION)
000724                           TO WS-PREV-1-DIGIT
000725                    ELSE
000726                        MOVE 0 TO WS-PREV-1-DIGIT
000727                    END-IF.
000728
000729                    IF WS-DIGIT-POSITION > 2 THEN
000730                        COMPUTE WS-PREV-POSITION = WS-DIGIT-POSITION - 2
000731                        MOVE WS-SALARY-DIGIT-ARRAY(WS-PREV-POSITION)
000732                           TO WS-PREV-2-DIGIT
000733                    ELSE
000734                        MOVE 0 TO WS-PREV-2-DIGIT
000735                    END-IF.
000736
000737              *-   START-IF TRANSLATE DIGIT INTO WORD BASED ON PLACE VALUE -*
000738                    IF ( (WS-DIGIT-POSITION = 2 ) OR
000739                         (WS-DIGIT-POSITION = 5 ) OR
000740                         (WS-DIGIT-POSITION = 8 )  )
000741              *-     ---- TRANSLATE TENS PLACE ---------------------------*
000742              *-          SKIP TRANSLATION IF CURRENT-DIGIT = 0
000743              *-          EVALUATE AS "TEENS" IF CURRENT-DIGIT = 1
000744                      IF ( (WS-CURRENT-DIGIT NOT = 0) AND
000745                           (WS-CURRENT-DIGIT NOT = 1)     )
000746                        PERFORM 0640-TRANSLATE-TENS THRU 0640-EXIT
000747                      END-IF
000748                    ELSE
000749
000750              *-     START-IF 'ONES' VS. 'TEENS' VS. 'HUNDREDS' ----------*
000751                      IF ( (WS-DIGIT-POSITION = 3 ) OR
000752                           (WS-DIGIT-POSITION = 6 ) OR
000753                           (WS-DIGIT-POSITION = 9 )  )
000754
000755              *-         ---- TRANSLATE AS TEENS -------------------------*
```

```
000756                        IF (WS-PREV-1-DIGIT = 1)
000757                          PERFORM 0630-TRANSLATE-TEENS THRU 0630-EXIT
000758                        ELSE
000759                   *-     ---- TRANSLATE AS ONES, SKIP IF DIGIT = 0 -------*
000760                          IF (WS-CURRENT-DIGIT NOT = 0)
000761                            PERFORM 0620-TRANSLATE-ONES THRU 0620-EXIT
000762                          END-IF
000763                        END-IF
000764                      ELSE
000765                   *-     ---- TRANSLATE HUNDREDS PLACE, SKIP IF DIGIT = 0 *
000766                        IF (WS-CURRENT-DIGIT NOT = 0)
000767                          PERFORM 0620-TRANSLATE-ONES THRU 0620-EXIT
000768                        END-IF
000769                      END-IF
000770                   *-   END-IF 'ONES' VS. 'TEENS' VS. 'HUNDREDS' ------------*
000771                    END-IF.
000772                   *-  END-IF TRANSLATE DIGIT INTO WORD BASED ON PLACE VALUE -*
000773
000774                   *- CHECK FOR 'MILLIONS' 'THOUSANDS' 'HUNDREDS' AS NEEDED ---*
000775                    IF ( (WS-DIGIT-POSITION = 1) OR
000776                         (WS-DIGIT-POSITION = 3) OR
000777                         (WS-DIGIT-POSITION = 4) OR
000778                         (WS-DIGIT-POSITION = 6) OR
000779                         (WS-DIGIT-POSITION = 7)   ) AND
000780                       (WS-CURRENT-DIGIT > 0)
000781                      PERFORM 0650-ADD-PLACENAMES THRU 0650-EXIT
000782                    ELSE
000783                      IF ( (WS-DIGIT-POSITION = 3) OR
000784                           (WS-DIGIT-POSITION = 6)   ) AND
000785                         ( (WS-PREV-1-DIGIT  > 0) OR
000786                           (WS-PREV-2-DIGIT  > 0)    )
000787                        PERFORM 0650-ADD-PLACENAMES THRU 0650-EXIT
000788                      END-IF
000789                    END-IF.
000790
000791                   *- BUMP WS-DIGIT-POSITION. ---------------------------------*
000792                    ADD 1 TO WS-DIGIT-POSITION.
000793
000794                    0610-EXIT.
000795                    EXIT.
000796                   *-------------------------------------------------------------*
000797                   *                                                             *
000798                   *                      TRANSLATE ONES                         *
000799                   *                                                             *
000800                   *-------------------------------------------------------------*
000801                    0620-TRANSLATE-ONES.
000802                    MOVE 0 TO WS-LOOP.
000803                    PERFORM WITH TEST AFTER
000804                      UNTIL WS-LOOP = WS-ONES-CHARS-IN-WORD(WS-DTABLE-OFFSET)
000805                        ADD 1 TO WS-LOOP
000806                        ADD 1 TO WS-LG-CURR-CHAR
000807                        MOVE WS-ONES-ARRAY(WS-DTABLE-OFFSET, WS-LOOP) TO
000808                          WS-LG-CHARS(WS-LG-CURR-CHAR)
000809                    END-PERFORM.
000810                    0620-EXIT.
000811                    EXIT.
000812                   *-------------------------------------------------------------*
000813                   *                                                             *
000814                   *                      TRANSLATE TEENS                        *
000815                   *                                                             *
000816                   *-------------------------------------------------------------*
000817                    0630-TRANSLATE-TEENS.
000818                    MOVE 0 TO WS-LOOP.
000819                    PERFORM WITH TEST AFTER
000820                      UNTIL WS-LOOP = WS-TEENS-CHARS-IN-WORD(WS-DTABLE-OFFSET)
000821                        ADD 1 TO WS-LOOP
000822                        ADD 1 TO WS-LG-CURR-CHAR
000823                        MOVE WS-TEENS-ARRAY(WS-DTABLE-OFFSET, WS-LOOP) TO
000824                          WS-LG-CHARS(WS-LG-CURR-CHAR)
000825                    END-PERFORM.
000826                    0630-EXIT.
000827                    EXIT.
000828                   *-------------------------------------------------------------*
000829                   *                                                             *
000830                   *                      TRANSLATE TENS                         *
000831                   *                                                             *
000832                   *-------------------------------------------------------------*
000833                    0640-TRANSLATE-TENS.
000834                    MOVE 0 TO WS-LOOP.
000835                    PERFORM WITH TEST AFTER
000836                      UNTIL WS-LOOP = WS-TENS-CHARS-IN-WORD(WS-DTABLE-OFFSET)
000837                        ADD 1 TO WS-LOOP
000838                        ADD 1 TO WS-LG-CURR-CHAR
000839                        MOVE WS-TENS-ARRAY(WS-DTABLE-OFFSET, WS-LOOP) TO
000840                          WS-LG-CHARS(WS-LG-CURR-CHAR)
```

```
000841                END-PERFORM.
000842             0640-EXIT.
000843                EXIT.
000844           *-------------------------------------------------------*
000845           *                                                       *
000846           *                ADD PLACE NAMES                        *
000847           *                                                       *
000848           *-------------------------------------------------------*
000849             0650-ADD-PLACENAMES.
000850                MOVE 0 TO WS-LOOP.
000851                PERFORM WITH TEST AFTER
000852                  UNTIL WS-LOOP = WS-PLACE-CHARS-IN-WORD(WS-PTABLE-OFFSET)
000853                    ADD 1 TO WS-LOOP
000854                    ADD 1 TO WS-LG-CURR-CHAR
000855                    MOVE WS-PLACE-ARRAY(WS-PTABLE-OFFSET, WS-LOOP) TO
000856                      WS-LG-CHARS(WS-LG-CURR-CHAR)
000857                END-PERFORM.
000858             0650-EXIT.
000859                EXIT.
000860           *-------------------------------------------------------*
000861           *                                                       *
000862           *                  WRITE TO OUTPUT FILES                *
000863           *                                                       *
000864           *-------------------------------------------------------*
000865             0700-WRITE-PAYCHECK-FILE.
000866
000867
000868                WRITE OUT-PAYCHECK-REC FROM WS-BLANKS
000869                   AFTER ADVANCING PAGE.
000870                WRITE OUT-PAYCHECK-REC FROM WS-LA-CHECK
000871                   AFTER ADVANCING 3 LINES.
000872                WRITE OUT-PAYCHECK-REC FROM WS-LB-CHECK
000873                WRITE OUT-PAYCHECK-REC FROM WS-LC-CHECK.
000874                WRITE OUT-PAYCHECK-REC FROM WS-LD-CHECK
000875                   AFTER ADVANCING 3 LINES.
000876                WRITE OUT-PAYCHECK-REC FROM WS-LE-CHECK.
000877                WRITE OUT-PAYCHECK-REC FROM WS-LF-CHECK.
000878                WRITE OUT-PAYCHECK-REC FROM WS-LG-CHECK.
000879                WRITE OUT-PAYCHECK-REC FROM WS-LH-CHECK.
000880                WRITE OUT-PAYCHECK-REC FROM WS-BLANKS
000881                   AFTER ADVANCING 3 LINES.
000882                WRITE OUT-PAYCHECK-REC FROM WS-LINEI-CHECK.
000883                WRITE OUT-PAYCHECK-REC FROM WS-LINEJ-CHECK
000884                   BEFORE ADVANCING 1 LINE.
000885
000886                IF (NOT WS-OUT-PAYCHECK-FILE-OK)
000887                  MOVE WS-ABEND-WRITE TO WS-ABEND-CODE
000888                  MOVE WS-ABEND-WRITE TO WS-ERR2-ABEND
000889                  MOVE WS-OUT-PAYCHECK-FILE-STAT TO WS-ERR2-STATUS
000890                  WRITE OUT-ERROR-REC FROM WS-ERR2-REC
000891                  PERFORM 0900-CLEANUP THRU 0900-EXIT
000892                  CALL 'ILBOABNO' USING WS-ABEND-CODE
000893                END-IF.
000894
000895             0700-EXIT.
000896                EXIT.
000897           *-------------------------------------------------------*
000898           *                                                       *
000899           * RE-WRITE UPDATED EMPLOYEE RECORD BACK TO MASTER FILE  *
000900           *                                                       *
000901           *-------------------------------------------------------*
000902             0800-REWRITE-EMPLOYEE-FILE.
000903
000904                REWRITE IO-EMPLOYEE-REC FROM WS-EMPLOYEE-REC
000905                   INVALID KEY
000906                        MOVE WS-ABEND-REWRITE-BAD-KEY TO WS-ABEND-CODE
000907                        MOVE WS-ABEND-REWRITE-BAD-KEY TO WS-ERR1-ABEND
000908                        MOVE IO-EMPLOYEE-KEY1-ENUM TO WS-ERR1-KEY
000909                        WRITE OUT-ERROR-REC FROM WS-ERR1-REC
000910                        PERFORM 0900-CLEANUP THRU 0900-EXIT
000911                        CALL 'ILBOABNO' USING WS-ABEND-CODE.
000912
000913                IF  (NOT WS-IO-EMPLOYEE-FILE-OK)
000914                  MOVE WS-ABEND-REWRITE TO WS-ABEND-CODE
000915                  MOVE WS-ABEND-REWRITE TO WS-ERR1-ABEND
000916                  MOVE WS-IO-EMPLOYEE-FILE-STAT TO WS-ERR2-STATUS
000917                  WRITE OUT-ERROR-REC FROM WS-ERR2-REC
000918                  PERFORM 0900-CLEANUP THRU 0900-EXIT
000919                  CALL 'ILBOABNO' USING WS-ABEND-CODE
000920                END-IF.
000921
000922             0800-EXIT.
000923                EXIT.
000924           *-------------------------------------------------------*
000925           *                                                       *
```

```
000926              *                      CLOSE FILES                      *
000927              *                                                       *
000928              *-------------------------------------------------------*
000929               0900-CLEANUP.
000930
000931                   CLOSE  IN-TIMESHEET-FILE
000932                          IO-EMPLOYEE-FILE
000933                          OUT-PAYCHECK-FILE
000934                          OUT-ERROR-FILE.
000935
000936                   IF ( (NOT WS-IN-TIMESHEET-FILE-OK)  OR
000937                        (NOT WS-IO-EMPLOYEE-FILE-OK )  OR
000938                        (NOT WS-OUT-PAYCHECK-FILE-OK)  OR
000939                        (NOT WS-OUT-ERROR-FILE-OK)        )
000940                      DISPLAY 'TIMESHEET = ' WS-IN-TIMESHEET-FILE-STAT
000941                              ', EMPLOYEE  = ' WS-IO-EMPLOYEE-FILE-STAT
000942                              ', PAYCHECK  = ' WS-OUT-PAYCHECK-FILE-STAT
000943                              ', ERROR     = ' WS-OUT-ERROR-FILE-STAT
000944                      MOVE 'Y' TO WS-DONE-SW
000945                   END-IF.
000946
000947               0900-EXIT.
000948                   EXIT.
.....               Alternate table initialization routine
000990               END PROGRAM PAYROLL0.
```

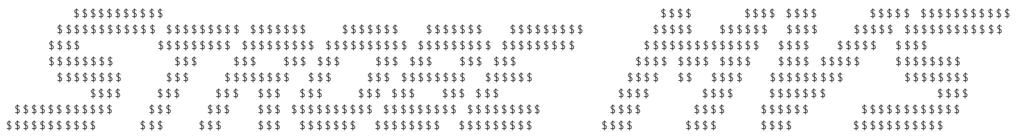
TAXRTN0

```
000001               IDENTIFICATION DIVISION.
000002
000003               PROGRAM-ID. TAXRTN0.
000004
000005               AUTHOR.   A. PROGRAMMER
000006
000007              ****************************************************************
*
000008              *
000009              *  S T R O B E     T E S T    P R O G R A M
000010              *
000011              ****************************************************************
*
000012              *                                    :
000013              *  MODULE......TAXRTN0 -             : THIS PROGRAM READS THE
000014              *                                    : TAXFILE FOR THE STATE'S
000015              *                                    : TAX RATE.
000016              *  DATE........1 JAN 1996            :
000017              *                                    :
000018              *                                    :
000019              *================================================================
=
000020              * DESCRIPTION:
000021              * THIS PROGRAM IS AN IN-HOUSE FRONT END FOR A TAX DATA SERVICE
000022              * PURCHASED FROM AN EXTERNAL VENDOR.   THE VENDOR SUPPLIES
000023              * VSAM FILES CONTAINING THE LATEST FEDERAL, STATE, AND LOCAL
000024              * INCOME TAX RATES.  NOTE:  TO SIMPLIFY THIS EXAMPLE, ONLY
000025              * STATE INCOME TAX IS SIMULATED.
000026              *
000027              * FILES:
000028              * THE PROGRAM REQUIRES THE FOLLOWING FILES:
000029              * 1. TAXDATA FILE: ONE RECORD FOR EACH STATE LISTING STATE
000030              *    ABBREVIATION AND INCOME TAX RATE.
000031              *
000032              * THE INVOKING ROUTINE SUPPLIES TWO PARAMETERS AS FOLLOWS:
000033              *
000034              * INPUT:
000035              * 1. LINKAGE-TAXFILE-STATE:  PASS IN STATE TO LOOKUP
000036              * 2. LINKAGE-TAXFILE-RATE:   UNINITIALIZED
000037              *
000038              * OUTPUT:
000039              * 1. LINKAGE-TAXFILE-STATE:  UNCHANGED
000040              * 2. LINKAGE-TAXFILE-RATE:   SET TO CORRECT TAX
000041              *
000042              *PROGRAM LOGIC FLOW:
000043              * -OPEN FILES.
000044              * -USING VSAM KEYED ACCESS, LOOKUP THE CORRECT STATE TAX RATE
000045              * -CLOSE FILES.
000046              *================================================================
=
000047              *
000048              *  INPUT FILES:   I-TAXFILE            VSAM FILE
```

```
000049              *
000050              *================================================================
=
000052               ENVIRONMENT DIVISION.
000053               CONFIGURATION SECTION.
000054               SOURCE-COMPUTER. IBM-370.
000055               OBJECT-COMPUTER. IBM-370.
000056               INPUT-OUTPUT SECTION.
000057               FILE-CONTROL.
000058
000059                   SELECT I-TAXFILE ASSIGN UT-DA-TAXDATA
000060                          ORGANIZATION IS INDEXED
000061                          ACCESS       IS DYNAMIC
000062                          RECORD KEY   IS I-TAXFILE-REC-KEY
000063                          FILE STATUS  IS WS-I-TAXFILE-STAT.
000064
000065               DATA DIVISION.
000066               FILE SECTION.
000067               FD  I-TAXFILE
000068                   BLOCK CONTAINS 0 RECORDS
000069                   LABEL RECORDS ARE STANDARD.
000070
000071               01  I-TAXFILE-REC.
000072                   05  I-TAXFILE-REC-KEY         PIC  X(02).
000073                   05  FILLER                    PIC  X(08).
000074
000075               WORKING-STORAGE SECTION.
000076
000077               77  WS-ABEND-CODE                 PIC  S9(4) COMP.
000078
000079               01  VARS.
000080
000081                   05  WS-PROGRAM-NAME           PIC  X(8) VALUE 'TAXRTN0 '.
000082                   05  WS-TAXFILE-CTR            PIC S9(9) VALUE +0 COMP-3.
000083
000084                   05  WS-OPEN-ABEND             PIC S9(4) VALUE +2001.
000085                   05  WS-INV-KEY-ABEND          PIC S9(4) VALUE +2002.
000086                   05  WS-FILE-ERROR-ABEND       PIC S9(4) VALUE +2003.
000087                   05  WS-CLOSE-ABEND            PIC S9(4) VALUE +2004.
000088
000089               01  WS-FILE-LAYOUTS.
000090
000091                   05  WS-TAXFILE-REC.
000092                       10  WS-TAXFILE-KEY        PIC  X(02).
000093                       10  WS-TAXFILE-RATE       PIC  V9(2).
000094                       10  FILLER                PIC  X(08).
000095
000096               01  WS-SWITCHES.
000097
000098                   05  WS-I-TAXFILE-STAT PIC          X(2).
000099                       88  WS-I-TAXFILE-OK                 VALUE '00'.
000100                       88  WS-I-TAXFILE-EOF                VALUE '10'.
000101                       88  WS-I-TAXFILE-BAD-RANDOM-KEY     VALUE '23'.
000102
000103
000104               LINKAGE SECTION.
000105               01  LINKAGE-PARMS.
000106                   02  LINKAGE-TAXFILE-STATE     PIC  X(02).
000107                   02  LINKAGE-TAXFILE-RATE      PIC  V9(2).
000108
000109              *-------------------------------------------------------------*
000110              *                                                             *
000111              *    TOP OF PROGRAM LOGIC                                     *
000112              *                                                             *
000113              *-------------------------------------------------------------*
000114               PROCEDURE DIVISION USING LINKAGE-PARMS.
000115
000116                   PERFORM 0100-OPEN
000117                      THRU 0100-EXIT.
000118                   PERFORM 0200-READ-TAXFILE
000119                      THRU 0200-EXIT.
000120                   PERFORM 0300-CLOSE
000121                      THRU 0300-EXIT.
000122
000123                   GOBACK.
000124              *-------------------------------------------------------------*
000125              *                                                             *
000126              *            OPEN FILE FOR INPUT.                             *
000127              *                                                             *
000128              *-------------------------------------------------------------*
000129               0100-OPEN.
000130
000131                   OPEN INPUT  I-TAXFILE.
000132
000133                   IF WS-I-TAXFILE-OK
```

```
000134                    GO TO 0100-EXIT.
000135
000136                MOVE WS-OPEN-ABEND TO WS-ABEND-CODE
000137                CALL 'ILBOABNO' USING WS-ABEND-CODE.
000138
000139           0100-EXIT.
000140                EXIT.
000141          *---------------------------------------------------------*
000142          *                                                         *
000143          *                   READ TAX DATA FILE                    *
000144          *                                                         *
000145          *---------------------------------------------------------*
000146           0200-READ-TAXFILE.
000147
000148                MOVE LINKAGE-TAXFILE-STATE TO I-TAXFILE-REC-KEY.
000149
000150                READ I-TAXFILE INTO WS-TAXFILE-REC
000151                   KEY IS I-TAXFILE-REC-KEY.
000152
000153                IF  WS-I-TAXFILE-OK
000154                    MOVE WS-TAXFILE-RATE TO LINKAGE-TAXFILE-RATE
000155                    GO TO 0200-EXIT.
000156
000157                IF  WS-I-TAXFILE-BAD-RANDOM-KEY
000158                    MOVE ZEROS TO LINKAGE-TAXFILE-RATE
000159                    MOVE WS-INV-KEY-ABEND TO WS-ABEND-CODE
000160                    CALL 'ILBOABNO' USING WS-ABEND-CODE
000161
000162                ELSE
000163                    MOVE WS-FILE-ERROR-ABEND TO WS-ABEND-CODE
000164                    CALL 'ILBOABNO' USING WS-ABEND-CODE.
000165
000166           0200-EXIT.
000167                EXIT.
000168          *---------------------------------------------------------*
000169          *                                                         *
000170          *                      CLOSE FILES                        *
000171          *                                                         *
000172          *---------------------------------------------------------*
000173           0300-CLOSE.
000174
000175                CLOSE  I-TAXFILE.
000176
000177                IF WS-I-TAXFILE-OK
000178                   GO TO 0300-EXIT.
000179
000180                MOVE WS-CLOSE-ABEND TO WS-ABEND-CODE
000181                CALL 'ILBOABNO' USING WS-ABEND-CODE.
000182
000183           0300-EXIT.
000184                EXIT.
000185           END PROGRAM TAXRTNO.
```

Sample STROBE Performance Profile for PAYROLL0



```
                                    ** MEASUREMENT SESSION DATA **

------- JOB ENVIRONMENT --------        ------- MEASUREMENT PARAMETERS ------        ----- MEASUREMENT STATISTICS ----

PROGRAM MEASURED   -    PAYROLL0        ESTIMATED SESSION TIME  -      45 MIN        CPS TIME PERCENT -          10.51
JOB NAME           -    WPAKMOG0        TARGET SAMPLE SIZE      -      10,000        WAIT TIME PERCENT -         89.49
JOB NUMBER         -    JOB01541        REQUEST NUMBER   (Q)    -          36        RUN MARGIN OF ERROR PCT  -    .97
STEP NAME   -             RUNAPP                                                    CPU MARGIN OF ERROR PCT  -   2.99

DATE OF SESSION    -  05/12/1998        SYS REQ - ALL SYSTEMS                       TOTAL SAMPLES TAKEN -      10,248
TIME OF SESSION    -    11:25:53                                                    TOTAL SAMPLES PROCESSED -  10,248
CONDITION CODE     -      C-0000        MODULE MAPPING BASELINE -           2       INITIAL SAMPLING RATE-   3.70/SEC
                                        BASELINE OVERRIDE -           TAXRTN0       FINAL SAMPLING RATE  -   3.70/SEC
SYSTEM -             ESA SP5.2.0        LIBRARY -            WPAKMO.USER.LOAD
DFSMS              -       1.2.0                                                   SESSION TIME -   52 MIN  5.98 SEC
CPU MODEL          -   3090-600S                                                   CPU TIME -        5 MIN 35.50 SEC
SMF/SYSTEM ID -       MVST/TLP01        --------- REPORT PARAMETERS ---------      WAIT TIME    -   45 MIN 53.92 SEC
                                                                                  STRETCH TIME -    0 MIN 36.56 SEC
REGION SIZE BELOW 16M  -  4,160K        REPORT RESOLUTION    -       64 BYTES
REGION SIZE ABOVE  -     32,768K        SORTSIZE             -      999,999        SRB TIME     -    0 MIN 17.40 SEC
                                        LINES/PAGE           -           60        SERVICE UNITS-            334494
PTF LVL- 2.2.0.FS000000/FS000000
STROBE TAPE NUMBER -  000-S00DSK        DASD= 2.0%  DASDGAP= 5                    PAGES IN-      0     OUT-       0
                                                                                  PAGING RATE  -           0.00/SEC
                                                                                  EXCPS -     38,093      12.19/SEC
                                        DATE FORMAT               MM/DD/YYYY
                                        TIME FORMAT (24 HOURS)      HH:MM:SS

SAMPLE DATA SET  - STROBE.WPAJEA.S001D001

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

STROBE* PERFORMANCE PROFILE                          PAYROLL0                        06/15/1998          PAGE   2

                                  ** TIME DISTRIBUTION OF ACTIVITY LEVEL **

TASK OR    RESOURCE  N X 10 PLUS OR MINUS 5 IS PERCENT OF FULL UTILIZATION   * IS GREATER THAN 95%     - IS LESS THAN 5%
 DDNAME             .-------------------------------------------------------------------------------------------------.
PAYROLL0   CPU      .-   -    ------ -- ------------ ---------- ---- --- ----- -  ----------- ----- -------- -  --- --- -.
.FILEMGT            .1111111111121112111211111111112221111121111222121111-11-1111-1111111112111111111111111-111-12111211.
PAYCHECK   3380     .--- --11-1---11-----1---1---- ---1--------111--- ---------  --1-  ----------1----1-1-- --- -- --------.
TAXDATA    3380     .-- ---------   ---- 1-- -- ---- ----11--------1- ----------------------1--1---------1-- ------ -- - -.
EMPLOYEE   3380     .----------------- -------- ------ ---1----1- -------- ------------  -------------- ----------1---- 1.
SYS00010   3380     .  -  -   --        -                                                 -                          .
                    0----0----1----1----2----2----3----3----4----4----5----5----6----6----7----7----8----8----9----9----*.
                    0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----*.
                    START RUN                PERCENT OF ALLOCATED RUN TIME                                  END RUN

                                  ** RESOURCE DEMAND DISTRIBUTION **

                         ---- PERCENT OF RUN TIME -----    ----- PERCENT OF RUN TIME SPENT -------   CUMULATIVE PERCENTAGES
TASK OR    RESOURCE     SERVICED  SERVICED  SERVICED      SOLO     SOLO     SOLO     CAUSING         SOLO     CAUSING
 DDNAME                  BY CPU    BY I/O  BY EITHER     IN CPU   IN I/O  IN EITHER   CPU WAIT        TIME    CPU WAIT

PAYROLL0   CPU             1.22       .00      1.22       1.21      .00      1.21        .24         1.21        .24
.FILEMGT                   8.58       .89      9.46       8.57      .88      9.46      80.47        10.67      80.71
PAYCHECK   3380             .00      2.74      2.74        .00     2.74      2.74       2.98        13.41      83.69
TAXDATA    3380             .67      1.42      2.10        .67     1.41      2.09       2.40        15.50      86.09
EMPLOYEE   3380             .04      1.95      1.99        .04     1.95      1.99       3.33        17.49      89.42
SYS00010   3380             .00       .07       .07        .00      .07       .07        .07        17.56      89.49

                                  ** WORKING SET SIZE THROUGH TIME **

                    .-------------------------------------------------------------------------------------------------.
                600 .                                                                                                 .
                    .                                                                                                 .
                    .                                                                                                 .
 AVERAGE            .                                                                                                 .
 PAGE FRAME         .+                                                                                                .
 COUNT          300 .+                                                                                                .
                    .+                                                                                                .
                    .++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++.
                    .                                                                                                 .
                    .                                                                                                 .
                  0 .                                                                                                 .
                    0----0----1----1----2----2----3----3----4----4----5----5----6----6----7----7----8----8----9----9----*.
                    0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----*.
                    START RUN                PERCENT OF ALLOCATED RUN TIME                                  END RUN

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

STROBE* PERFORMANCE PROFILE                          PAYROLL0                        06/15/1998          PAGE   3
```

** WAIT TIME BY MODULE **

```
MODULE   SECTION COMPRESSED   FUNCTION                RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:   .97%
 NAME      NAME    SECTION                             PAGE     TOTAL        .00     14.00     28.00     42.00     56.00

.IOCS    IGG019AR          QSAM PUT  NEXT BUFFER        .00      2.98        .++

.SVC     SVC 008           PROGRAM MANAGER/LOAD         .00       .01        .
.SVC     SVC 019           OPEN                         .00      4.00        .++
.SVC     SVC 020           CLOSE                        .00       .66        .
.SVC     SVC 026           CATALOG MANAGEMENT           .00     55.36        .++++++++++++++++++++++++++++++++++++++++
.SVC     SVC 056           RESOURCE MANAGER/ENQUEUE     .00     20.52        .+++++++++++++++
.SVC     SVC 130           RACHECK                      .00       .23        .
                                                       -----     -----
.SVC     TOTALS            SUPERVISOR CONTROL           .00     80.78

.VSAM    IDA019L1          VSAM RECORD MANAGEMENT       .00      5.73        .++++
                                                       -----     -----
PROGRAM PAYROLLO TOTALS                                 .00     89.49
```

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

** DATA SET CHARACTERISTICS **

```
DDNAME     ACCESS    POOL  REC  BLK/CI    HBUF  BUF  RPL  -SPLITS-     EXCP DATA SET NAME
           METHOD      NO  SIZE  SIZE       NO   NO STRNO CI    CA   COUNTS

EMPLOYEE VSAM KSDS          150  4096              2   1               7200 WPAV.WPAKMO.PAY01.EMPLOYEE
EMPLOYEE VSAM INDX         4089  4096              1                  13875 WPAV.WPAKMO.PAY01.EMPLOYEE
PAYCHECK QSAM               132  1320              5                  16800 WPAKMO.PAY01.OUTPUT.PAYCHEKS
STEPLIB                                                                   6
TAXDATA  VSAM KSDS           10  4096              2   1                 67 WPAV.WPAKMO.PAY01.TAXDATA
TAXDATA  VSAM INDX         4089  4096              1                    145 WPAV.WPAKMO.PAY01.TAXDATA
```

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

** DATA SET CHARACTERISTICS SUPPLEMENT **

```
DDNAME   ACCESS     DSNAME                                    OPEN INTENT\PROCESSING MODE
         METHOD

EMPLOYEE VSAM KSDS  WPAV.WPAKMO.PAY01.EMPLOYEE                OUTPUT,DIR
  FILE:  RECFM.....                                          VSAM:  FREESPACE... 916K    USER RECORDS...         9,999
         EXTENTS.....  1                                           SHROPTS....(1,3)     LOGICAL OPERATIONS:
                                                                   CI/CA.....  150       DELETES........             0
                                                                   %CI FREE.....  0      UPDATES........         6,937
                                                                   %CA FREE.....  0      RETRIEVES......         6,938
                                                                                        INSERTS........             0

EMPLOYEE VSAM INDEX WPAV.WPAKMO.PAY01.EMPLOYEE                OUTPUT,DIR
  FILE:  RECFM.....                                          VSAM:  FREESPACE...  24K    USER RECORDS...             4
         EXTENTS.....  1                                           SHROPTS....(1,3)     LOGICAL OPERATIONS:
                                                                   CI/CA.....   10       DELETES........             0
                                                                   %CI FREE.....  0      UPDATES........             0
                                                                   %CA FREE.....  0      RETRIEVES......             0
                                                                   INDEX LVLS.    2      INSERTS........             0

PAYCHECK QSAM       WPAKMO.PAY01.OUTPUT.PAYCHEKS              OUTPUT
  FILE:  RECFM.....FBM
         EXTENTS.....  4


TAXDATA  VSAM KSDS  WPAV.WPAKMO.PAY01.TAXDATA                 INPUT,DYN
  FILE:  RECFM.....                                          VSAM:  FREESPACE... 396K    USER RECORDS...            58
         EXTENTS.....  1                                           SHROPTS....(1,3)     LOGICAL OPERATIONS:
                                                                   CI/CA.....  100       DELETES........             0
                                                                   %CI FREE.....  0      UPDATES........             0
                                                                   %CA FREE.....  0      RETRIEVES......             0
                                                                                        INSERTS........             0

TAXDATA  VSAM INDEX WPAV.WPAKMO.PAY01.TAXDATA                 INPUT,DYN
  FILE:  RECFM.....                                          VSAM:  FREESPACE...  36K    USER RECORDS...             1
         EXTENTS.....  1                                           SHROPTS....(1,3)     LOGICAL OPERATIONS:
                                                                   CI/CA.....   10       DELETES........             0
                                                                   %CI FREE.....  0      UPDATES........             0
                                                                   %CA FREE.....  0      RETRIEVES......             0
                                                                   INDEX LVLS.    1      INSERTS........             0
```

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

** I/O FACILITY UTILIZATION SUMMARY **

```
 UNIT     DEVICE     CACHE  VOLUME DDNAME             RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:   .97%
  NO       TYPE      ELIG    ID                 I/O   SOLO     TOTAL        .00      1.00      2.00      3.00      4.00

  923  DA 3380K             WPA001 .FILEMGT            .87       .88        .********
  923  DA 3380K             WPA001 EMPLOYEE       I    .02       .02        .
  923  DA 3380K             WPA001 EMPLOYEE       O    .75       .75        .*******
  923  DA 3380K             WPA001 EMPLOYEE INDEX I   1.18      1.18        .***********
  923  DA 3380K             WPA001 SYS00010            .07       .07        .
  923  DA 3380K             WPA001 TAXDATA        I    .65       .65        .******
  923  DA 3380K             WPA001 TAXDATA  INDEX I    .76       .77        .*******
                                                      -----     -----
```

```
UNIT   923 TOTALS     C&DFW                          4.30      4.32

  92A  DA 3380K             WPA008 .FILEMGT          .01       .01        .
  92A  DA 3380K             WPA008 PAYCHECK       0  2.74      2.74        .**************************
                                                    -----     -----
UNIT   92A TOTALS     C&DFW                          2.75      2.75
```

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

```
STROBE* PERFORMANCE PROFILE                          PAYROLLO                          06/15/1998          PAGE   7

                                        ** MOST INTENSIVELY EXECUTED PROCEDURES **

MODULE    SECTION    LINE     PROCEDURE/FUNCTION          STARTING   PROCEDURE   CPU TIME PERCENT   CUMULATIVE PERCENT
 NAME       NAME    NUMBER        NAME                    LOCATION    LENGTH     SOLO     TOTAL      SOLO     TOTAL

.SVC      SVC 056                  RESOURCE MANAGER/ENQUEUE                     34.35     34.35     34.35     34.35
.SVC      SVC 019                  OPEN                                         22.10     22.19     56.45     56.54
.SVC      SVC 020                  CLOSE                                        16.53     16.53     72.98     73.07
.SVC      SVC 026                  CATALOG MANAGEMENT                            8.54      8.54     81.52     81.61
.VSAM     IDA019RO                 VSAM R/M INTERFACE                            6.31      6.31     87.83     87.92
.SVC      SVC 130                  RACHECK                                       5.57      5.66     93.40     93.58
.SVC      SVC 083                  SMF                                           1.76      1.76     95.16     95.34
.COBLIB   IGZCPCO          IGZESPM  SPACE MANAGER                                 1.39      1.39     96.55     96.73
.SVC      SVC 060                  ESTAE                                          .93       .93     97.48     97.66
.SVC      SVC 048                  RESOURCE MANAGER/DEQUEUE                       .56       .56     98.04     98.22
```

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

```
STROBE* PERFORMANCE PROFILE                          PAYROLLO                          06/15/1998          PAGE   8

                                        ** PROGRAM SECTION USAGE SUMMARY **

 MODULE  SECTION 16M   SECT FUNCTION                 CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.99%
   NAME  NAME    <,>   SIZE                          SOLO     TOTAL        .00     23.00     46.00     69.00     92.00

.SYSTEM  .COBLIB           COBOL LIBRARY SUBROUTINE   1.95      1.95        .
.SYSTEM  .IOCS             DATA MANAGEMENT SERVICES    .19       .19        .
.SYSTEM  .SVC              SUPERVISOR CONTROL        90.53     90.71        .****************************************
.SYSTEM  .VSAM             VIRTUAL STORAGE ACC METH   6.78      6.78        .**
                                                      -----     -----
.SYSTEM  TOTALS            SYSTEM SERVICES           99.45     99.63

PAYROLLO PAYROLLO <   14680                            .37       .37        .
                                                      -----     -----
PROGRAM PAYROLLO TOTALS                               99.81    100.00
```

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

```
STROBE* PERFORMANCE PROFILE                          PAYROLLO                          06/15/1998          PAGE   9

                                        ** PROGRAM USAGE BY PROCEDURE **

        .SYSTEM     SYSTEM SERVICES                    .COBLIB   COBOL LIBRARY SUBROUTINE

MODULE   SECTION        FUNCTION           INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.99%
  NAME      NAME                            LENGTH   SOLO     TOTAL        .00       .50      1.00      1.50      2.00

IGZCPAC  IGZCSPA  PRINTER SPACING             2428    .19       .19        .***
IGZCPAC  IGZCVLD  VERIFY LOADER               1700    .09       .09        .*
IGZCPCO  IGZESPM  SPACE MANAGER               3632   1.39      1.39        .**************************
IGZCPCO  IGZEVIO  VSAM INPUT/OUTPUT           12128    .09       .09        .*
IGZCPCO  IGZEVOC  VSAM INTERFACE AND CLOSE     4008    .09       .09        .*
IGZCPCO  IGZEVOP  VSAM OPEN                   12152    .09       .09        .*
                                                      -----     -----
        .COBLIB   TOTALS                              1.94      1.94

        .SYSTEM     SYSTEM SERVICES                    .IOCS     DATA MANAGEMENT SERVICES

MODULE   SECTION        FUNCTION           INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.99%
  NAME      NAME                            LENGTH   SOLO     TOTAL        .00       .50      1.00      1.50      2.00

IECOSCR1          PHYSICAL IOCS                3424    .19       .19        .***
                                                      -----     -----
        .IOCS     TOTALS                              .19       .19

        .SYSTEM     SYSTEM SERVICES                    .SVC      SUPERVISOR CONTROL

MODULE   SECTION        FUNCTION           INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.99%
  NAME      NAME                            LENGTH   SOLO     TOTAL        .00      9.00     18.00     27.00     36.00

SVC 019           OPEN                             22.10     22.19        .************************
SVC 020           CLOSE                            16.53     16.53        .******************
```

```
SVC 026          CATALOG MANAGEMENT                  8.54      8.54        .*********
SVC 040          TASK MANAGER/EXTRACT                 .09       .09        .
SVC 048          RESOURCE MANAGER/DEQUEUE             .56       .56        .
SVC 056          RESOURCE MANAGER/ENQUEUE           34.35     34.35        .**************************************
SVC 060          ESTAE                                .93       .93        .*
SVC 083          SMF                                 1.76      1.76        .*
SVC 117          DEB VALIDITY CHECKING                .09       .09        .
SVC 130          RACHECK                             5.57      5.66        .******
                                                     -----     -----
         .SVC    TOTALS                             90.52     90.70
```

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

```
STROBE* PERFORMANCE PROFILE                          PAYROLLO                          06/15/1998          PAGE  10

                                          ** PROGRAM USAGE BY PROCEDURE **

         .SYSTEM     SYSTEM SERVICES                    .VSAM     VIRTUAL STORAGE ACC METH

MODULE   SECTION        FUNCTION           INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.99%
  NAME     NAME                             LENGTH   SOLO     TOTAL        .00      2.00      4.00      6.00      8.00

IDA019L1         VSAM RECORD MANAGEMENT     214384    .46       .46        .**
IDA019RO         VSAM R/M INTERFACE           3096   6.31      6.31        .********************************
                                                     -----     -----
         .VSAM   TOTALS                              6.77      6.77
```

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

```
STROBE* PERFORMANCE PROFILE                          PAYROLLO                          06/15/1998          PAGE  11

                                          ** PROGRAM USAGE BY PROCEDURE **

MODULE - PAYROLLO                                    SECTION - PAYROLLO
                                                     SOURCE LANGUAGE - IBM COBOL II                                VC40X26

 LINE     PROCEDURE          STARTING       INTERVAL  CPU TIME PERCENT       CPU TIME HISTOGRAM  MARGIN OF ERROR:  2.99%
NUMBER      NAME            LOCATION         LENGTH   SOLO     TOTAL        .00       .50      1.00      1.50      2.00

        DATA DIVISION         000000           6566    .00       .00      - .
   837     ADD                0019A6             82    .28       .28        .*****
   842  0640-EXIT             0019F8            156    .00       .00      - .
   865  0700-WRIT..HECK-FILE  001A94             94    .09       .09        .*
   872     WRITE              001AF2           7782    .00       .00      - .
                                                     -----     -----
SECTION  PAYROLLO  TOTALS                             .37       .37
```

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

```
STROBE* PERFORMANCE PROFILE                          PAYROLLO                          06/15/1998          PAGE  12

                                          ** DASD USAGE BY CYLINDER **
DEVICE ADDRESS  -  923   TYPE - 3380

  VOLUME          DDNAME            CYLINDER          RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:   .97%
    ID                                NUMBER           SOLO     TOTAL        .00       .50      1.00      1.50      2.00

  WPA001         .FILEMGT              0       VTOC    .87       .88        .*****************

                 EMPLOYEE            635               .77       .77        .***************
                 EMPLOYEE INDEX       22              1.17      1.17        .************************
                 EMPLOYEE INDEX      635               .01       .01        .

                 SYS00010              0       VTOC    .07       .07        .*

                 TAXDATA              99               .65       .65        .*************
                 TAXDATA  INDEX        7               .76       .77        .***************
                                                     -----     -----
DEVICE ADDRESS  -  923 TOTALS                        4.30      4.32
```

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

```
STROBE* PERFORMANCE PROFILE                          PAYROLLO                          06/15/1998          PAGE  13

                                          ** DASD USAGE BY CYLINDER **
DEVICE ADDRESS  -  92A   TYPE - 3380

  VOLUME          DDNAME            CYLINDER          RUN TIME PERCENT       RUN TIME HISTOGRAM  MARGIN OF ERROR:   .97%
    ID                                NUMBER           SOLO     TOTAL        .00       .50      1.00      1.50      2.00

  WPA008         .FILEMGT            1569               .01       .01        .

                 PAYCHECK            1560-1572          .92       .92        .
```

```
                PAYCHECK            1752-1767          .98       .98        .
                PAYCHECK            2038-2050          .78       .78        .
                PAYCHECK            2505               .07       .07        .*
                                                      -----     -----
DEVICE ADDRESS  -  92A TOTALS                         2.75      2.75
```

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

STROBE* PERFORMANCE PROFILE PAYROLLO 06/15/1998 PAGE 14

** ATTRIBUTION OF CPU EXECUTION TIME **

.SVC SVC 019 OPEN

XACTION	MODULE	SECTION	RETURN	LINE	PROCEDURE NAME	MODULE	SECTION	FUNCTION	CPU TIME % SOLO	TOTAL
	TAXRTNO	TAXRTNO	000380	131	OPEN	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	22.10	22.19
									22.10	22.19

.SVC SVC 020 CLOSE

XACTION	MODULE	SECTION	RETURN	LINE	PROCEDURE NAME	MODULE	SECTION	FUNCTION	CPU TIME % SOLO	TOTAL
	PAYROLLO	PAYROLLO	001F00	931	CLOSE	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	.09	.09
	TAXRTNO	TAXRTNO	0004BA	175	CLOSE	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	16.43	16.43
									16.53	16.53

.SVC SVC 026 CATALOG MANAGEMENT

XACTION	MODULE	SECTION	RETURN	LINE	PROCEDURE NAME	MODULE	SECTION	FUNCTION	CPU TIME % SOLO	TOTAL
	TAXRTNO	TAXRTNO	000380	131	OPEN	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	6.31	6.31
	TAXRTNO	TAXRTNO	0004BA	175	CLOSE	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	2.23	2.23
									8.54	8.54

.SVC SVC 056 RESOURCE MANAGER/ENQUEUE

XACTION	MODULE	SECTION	RETURN	LINE	PROCEDURE NAME	MODULE	SECTION	FUNCTION	CPU TIME % SOLO	TOTAL
	TAXRTNO	TAXRTNO	000380	131	OPEN	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	26.37	26.37
	TAXRTNO	TAXRTNO	0004BA	175	CLOSE	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	7.99	7.99
									34.35	34.35

.SVC SVC 130 RACHECK

XACTION	MODULE	SECTION	RETURN	LINE	PROCEDURE NAME	MODULE	SECTION	FUNCTION	CPU TIME % SOLO	TOTAL
	TAXRTNO	TAXRTNO	000380	131	OPEN	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	5.57	5.66
									5.57	5.66

*STROBE (R) IS LICENSED BY COMPUWARE FOR USE BY COMPUWARE

STROBE* PERFORMANCE PROFILE PAYROLLO 06/15/1998 PAGE 15

** ATTRIBUTION OF CPU WAIT TIME **

.SVC SVC 019 OPEN

XACTION	MODULE	SECTION	RETURN	LINE	PROCEDURE NAME	MODULE	SECTION	FUNCTION	WAIT TIME % PAGE	TOTAL
	TAXRTNO	TAXRTNO	000380	131	OPEN	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	.00	4.00
									.00	4.00

.SVC SVC 026 CATALOG MANAGEMENT

XACTION	MODULE	SECTION	RETURN	LINE	PROCEDURE NAME	MODULE	SECTION	FUNCTION	WAIT TIME % PAGE	TOTAL
	.COBLIB	IGZCPCO	IGZEVOP	VSAM OPEN		SVC 019		OPEN	.00	.01
	PAYROLLO	PAYROLLO	000E6A	458	OPEN	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	.00	.02
	TAXRTNO	TAXRTNO	000380	131	OPEN	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	.00	34.71
	TAXRTNO	TAXRTNO	0004BA	175	CLOSE	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	.00	20.62
									.00	55.36

.SVC SVC 056 RESOURCE MANAGER/ENQUEUE

XACTION	MODULE	SECTION	RETURN	LINE	PROCEDURE NAME	MODULE	SECTION	FUNCTION	WAIT TIME % PAGE	TOTAL
	.COBLIB	IGZCPCO	IGZEVOP	VSAM OPEN		SVC 019		OPEN	.00	.01
	TAXRTNO	TAXRTNO	000380	131	OPEN	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	.00	17.17
	TAXRTNO	TAXRTNO	0004BA	175	CLOSE	IGZCPCO	IGZEVOC	VSAM INTERFACE AND CLOSE	.00	3.34
									.00	20.52

*STROBE IS A REGISTERED TRADEMARK OF COMPUWARE CORPORATION

Glossary

active request. A measurement request that initiates a measurement session to measure the address space of a job that is currently executing. (*See also* completed request, deferred request, measurement request, queued request, suspended request.)

ADABAS Feature. *See* STROBE ADABAS/NATURAL Feature.

APM. application performance management.

APMpower. The APMpower Application Performance Analysis System. Extends the benefits of STROBE to application developers who use workstations to develop, test, and maintain MVS applications. Developers employ the APMpower graphical user interface and advanced analytical aids to navigate the Performance Profile, analyze and improve application performance, and share performance knowledge across the IS organization. (*See also* STROBE.)

APMpower SQL Analysis Feature. Feature of APMpower that provides DB2 access-path explanations for resource-consumptive SQL (Structured Query Language) statements and offers rules-based help that points out opportunities to improve the performance of SQL statements.

application performance management (APM). A quality management discipline that enables IS organizations to deliver efficient and responsive applications, and maintain high standards of application performance throughout an application's life cycle.

attribution. A STROBE capability that relates CPU time or wait time within system service routines to the lines of code that invoked the routine. STROBE supports attribution for COBOL, PL/I, DB2, CICS, IMS, CA-IDMS, ADABAS, NATURAL, CSP, and IEF.

attribution reports. A series of reports that identify the sites where selected service routines were invoked. STROBE produces attribution reports for both CPU time and wait time.

CA-IDMS Feature. *See* STROBE CA-IDMS Feature.

CICS Feature. *See* STROBE CICS Feature.

codeblock. A portion of a control section whose size, in bytes, is equal to the report resolution specified by the STROBE user. In the STROBE Performance Profile, STROBE reports activity within control sections by codeblock.

completed request. A request associated with a measurement session that has ended. *(See also* active request, deferred request, measurement request, queued request, suspended request.)

Composer/IEF Feature. *See* STROBE COOL:Gen Feature.

concurrent set. A collection of STROBE measurement requests, stored in a request group, that enables measurement of multiple, related address spaces at the same time. A concurrent set must contain one trigger request and one or more related requests. (*See also* trigger request, related request.)

control data set. A partitioned data set into which an installation utility copies the contents of the STROBE installation tape.

control section. The part of a program specified by the programmer to be a relocatable unit, all elements of which are to be loaded into adjoining main storage locations.

COOL:Gen Feature. *See* STROBE COOL:Gen Feature.

CPS time percent. The percentage of time during which the central processing subsystem (comprising one or more CPUs) was in use by application tasks executing within the measured job step.

CPU time. The amount of time during the measurement session that one or more CPUs were executing tasks in the measured job step, exclusive of the measurement task itself, expressed in minutes and seconds to the nearest hundredth. (*See also* run time, total time.)

Cross-system Coupling Facility (XCF). A component available with MVS SP4.1 that provides functions to support cooperation and communication between authorized programs running within a sysplex.

CSP Feature. *See* STROBE CSP Feature.

data collector. *See* STROBE data collector and user-written data collector.

Data Facility Storage Management Subsystem (DFSMS). An IBM operating environment that helps automate and centralize the management of

storage for MVS environments. To manage storage, SMS provides the storage administrator with control over the data class, storage class, management class, storage group, and automatic class section routine devices.

DB2 Feature. *See* STROBE DB2 Feature.

deferred request. A measurement request that has not yet entered the active or queued state. A request has a status of deferred when it is a scheduled request or is a related request in a concurrent set. *(See also* active request, completed request, measurement request, queued request, suspended request, concurrent set.*)*

DFSMS. Data Facility Storage Management Subsystem.

execution sample. A sample in which STROBE detected CPU execution.

final session action. The action STROBE takes when the specified target sample size is reached (for example, CONTINUE or QUIT).

function descriptor. A short description of a module, pseudo-entity, or transaction's function.

history data set. The data set produced at installation, written to by the Reporter, and read by STROBE/ISPF and APMpower. The history data set contains historical information and calculated values for selected measurement sessions.

implantation. The process whereby STROBE adds the task structure and routines needed to perform measurements into the address space of the program to be measured.

IMS Feature. *See* STROBE IMS Feature.

Indexer. A STROBE program that uses a source language compiler output listing to locate procedures within a source module. Each STROBE Indexer accepts SYSPRINT data sets from a specific language compiler and produces a map data set that can be passed on to the STROBE Reporter. (*See also* indexing, module mapping, map data set.)

indexing. The process by which STROBE maps execution or wait activity to procedure names and line numbers. When a module has been indexed, the Performance Profile shows the module's line numbers and procedure names. (*See also* Indexer, module mapping, map data set.)

Interactive System Productivity facility (ISPF). An IBM interface used by STROBE that provides standard screen panels and interactive dialogs between the user and STROBE.

Interactive System Productivity Facility/Program Development Facility (ISPF/PDF). An ISPF facility that provides access to application development services for end-users and programmers.

Interface Feature. *See* STROBE Interface Feature.

ISPF. . *Interactive System Productivity Facility. (See also* STROBE/ISPF*)*

ISPF/PDF. . *Interactive System Productivity Facility/Program Development Facility.*

Java Feature. *See* STROBE Java Feature.

job. A collection of related programs identified by appropriate job control statements.

job step. A unit of work represented by a single program or procedure that contains a single program. A job consists of one or more job steps.

load module map. The basic control section structure of a load module. STROBE gathers this information from the library from which the program was loaded.

log data set. A data set that stores STROBE diagnostic-related data such as messages and trace data.

map data set. The data set produced by an Indexer program and read by the Reporter program. The map data set contains data that relates programmer-assigned procedure names and compiler-assigned statement numbers to addresses within the compiler-produced object modules.

measurement parameters. Information a user supplies to STROBE with a measurement request. Measurement parameters govern the type and amount of performance data collected, such as session duration, target sample size, data collectors and STROBE features invoked, module mapping information, the number of measurement sessions, and the final session action.

measurement request. A STROBE command to measure a batch processing program or online subsystem application. (*See also* measurement session, measurement task.)

measurement session. An interval during which STROBE collects measurement data during the execution of a target job step. (*See also* measurement request, measurement task.)

measurement session history record. An entry in the history data set. A measurement session history record is one iteration of a measurement session for a unique job, step, and program combination.

measurement task. STROBE programs that gather measurement data. The measurement task periodically interrupts program execution to gather data about system resource usage and records the information in a sample data set. (*See also* measurement request, measurement session.)

message data set. A STROBE data set that stores STROBE messages generated during measurement sessions.

module mapping. The STROBE process of identifying basic program structure from load module libraries. STROBE gathers module mapping data from the library from which the target program is loaded. It uses this information to attribute the sample data collected to the appropriate modules and control sections. (*See also* load module map.)

MQSeries Feature. *See* STROBE MQSeries Feature.

NATURAL Feature. *See* STROBE ADABAS/NATURAL Feature.

Online Library. *See* STROBE and APMpower Online Library.

overhead activities. Activities performed by MVS on behalf of a program. Service routines and subsystem routines are examples of overhead activities.

parameter data set. A data set specified by the system programmer during installation that contains STROBE parameter values.

Performance Profile. *See* STROBE Performance Profile.

procedure. *See* STROBE *procedure*.

program temporary fix (PTF). A program "patch" applied by the system programmer and used to fix an identified product defect.

pseudo-activity. The collection of file management operations that STROBE combines under the name .FILEMGT in the STROBE Performance Profile reports. (*See also* pseudo-entity.)

pseudo-entity. A group of supervisory and service components of the operating system that STROBE combines to make reports more concise. STROBE groups the components by function, and gives each group a unique name beginning with the character ".". STROBE defines four classes of pseudo-entities: pseudo-activities, pseudo-modules, pseudo-sections, and pseudo-transactions.

pseudo-module. The collection of all observed supervisor functions. The observed resource usage is then attributed to component pseudo-sections such as .COBLIB, .DB2, etc. (*See also* pseudo-entity.)

pseudo-section. The collection of either dynamically linked load modules, such as .IOCS (data management modules), or compiler library subroutines that are actually control sections within the target program load module, such as .COBLIB (COBOL library routines). Pseudo-sections are subsets of either a real load module or the pseudo-module .SYSTEM. (*See also* pseudo-entity.)

pseudo-transaction. The collection of online subsystem overhead functions that STROBE cannot assign to any specific transaction. For example, the pseudo-transaction .CICS includes all CICS supervisory functions that STROBE could not assign to a user-defined transaction. (*See also* pseudo-entity.)

PTF. program temporary fix.

QRE. *queue request element*.

queue data set. A STROBE data set that stores and preserves information about all requests that STROBE has not yet processed. (*See also* queued request.)

queue request element (QRE). The representation of a measurement request in the queue data set.

queued request. A measurement request that "waits" for the target job to start before activating a measurement session of a job step's address space. *(See also* active request, completed request, deferred request, measurement request, suspended request.*)*

related request. One of the two types of requests in a *concurrent set*. The measurement session for a related request is initiated and terminated by the trigger request. (*See also* concurrent set, trigger request.)

Reporter. A STROBE program that analyzes, reduces, and presents measurement data collected by the measurement facility in a series of reports called the STROBE Performance Profile. (*See also* reporting facility, STROBE Performance Profile.)

reporting facility. A component of STROBE that includes the Reporter program and the Indexers. (*See also* Indexer, Reporter.)

request group. A reusable collection of measurement requests that have been saved together under one name. (*See also* request group element.)

request group data set. A STROBE data set that stores the request groups and their associated request group elements.

request group element. A measurement request that is a member of a request group. (*See also* request group.)

request number. A unique four-byte identifier that STROBE assigns to each measurement request.

request status. The current state of a measurement request. The five possible status types are active, suspended, queued, completed, and deferred.

If the request is...	*Then the measurement task...*
active	is currently placing samples in a sample data set
suspended	is not active
queued	has not yet begun executing
completed	has terminated and all sample data sets have been closed
deferred	is either scheduled or a related request in a concurrent set

(See also active request, completed request, deferred request, queued request, suspended request.*)*

run time. The amount of time during which STROBE was measuring. It includes CPU time, comprising one or more CPUs, and wait time.

sample data set. The data set into which the measurement task places performance information gathered during a measurement session. Each sample data set is used as the basis for one STROBE Performance Profile.

sampling. The process that STROBE employs to gather information about an executing job or job step. (*See also* measurement request, sampling rate.)

sampling rate. The frequency at which STROBE takes measurement samples. STROBE calculates the rate from the session duration and the target sample size (rate=size/duration).

sequence number. A unique three-byte identifier STROBE assigns to the elements in a request group.

session duration. The estimated length of time that STROBE will sample the target job step.

SMS. *Storage Management Subsystem. See* Data Facility Storage Management Subsystem (DFSMS).

solo time. The percentage of *total time* during which STROBE detected one and only one activity for the measured job step. (*See also* total time.)

SQL. Structured Query Language.

SRE. STROBE *Request Element*.

SSMF. STROBE *Session Management Facility*.

stretch time. The estimated amount of time that the CPU was unavailable to process programs executing in the measured address space because of demands made by higher-priority address spaces and by service request blocks.

STROBE. The STROBE Application Performance Measurement System. Enables application developers and technical services personnel to deliver and maintain efficient and responsive applications throughout the application life cycle. STROBE measures activity in MVS-based online and batch processing applications and produces the STROBE Performance Profile, a hierarchical series of reports that detail where and how time is spent during application execution. (*See also* APMpower.)

STROBE ADABAS/NATURAL Feature. A Compuware product that extends the functions of STROBE so that users can measure and evaluate three types of performance data: (1) applications developed and executing in a NATURAL environment, (2) the ADABAS database management system, and (3) ADABAS wait time caused by 3GL programs that access it.

STROBE Advanced Session Management Feature.

A Compuware product that extends the functions of STROBE so that users can measure multiple steps of a job, measure related address spaces concurrently, create and submit groups of measurement requests, select global search characters, and submit measurement requests according to a schedule.

STROBE CA-IDMS Feature. A Compuware product that extends the functions of STROBE so that users can measure CA-IDMS applications.

STROBE CICS Feature. A Compuware product that extends the functions of STROBE for users of CICS applications. The STROBE CICS Feature enables users to produce a STROBE Performance Profile that shows activity in CICS transaction-processing programs, CICS supervisory and service programs, and data sets used by the CICS region.

STROBE command language. A set of commands issued by a user to STROBE through TSO, a batch job, or an operator's console. Commands allow users to add, change, and delete requests, and to control sampling in active measurement sessions.

STROBE COOL:Gen Feature. A Compuware product that extends the functions of STROBE so that users can measure applications created with Sterling Software's COOL:Gen product (formerly Texas Instruments' Composer product and Information Engineering Facility (IEF) product).

STROBE CSP Feature. A Compuware product that extends the functions of STROBE so that users can measure applications that were created with IBM's Cross System Product/370 (CSP/370).

STROBE data collector. A component of STROBE and its features that collects environment-specific performance data during a measurement session. (*See also* user-written data collector.)

STROBE DB2 Feature. A Compuware product that extends the functions of STROBE so that users can measure and evaluate the performance of applications that use IBM DATABASE 2 (DB2).

STROBE IMS Feature. A Compuware product that extends the functions of STROBE so that users can measure IMS DB- and DC-dependent regions and evaluate the performance of online and batch DL/I applications. In addition, users can measure and evaluate the performance of IMS system modules in the control region and DL/I Separate Address Space.

STROBE Interface Feature. A Compuware product that extends the functions of STROBE to users by providing interface support for user-written data collector programs.

STROBE/ISPF. A panel-driven interface that users employ to communicate with STROBE, for example, to submit measurement session requests or to create the STROBE Performance Profile.

STROBE Java Feature. A Compuware product that extends the functions of STROBE so that users can measure applications written in IBM's VisualAge for Java Enterprise Edition for OS/390. The Performance Profile shows the execution of Java methods, while new reports show CPU activity and class size for classes and their respective modules present during the measurement session, and CPU activity for Java methods within the classes and modules present during the measurement session.

STROBE MQSeries Feature. A Compuware product that extends the functions of STROBE so that users can measure applications that use IBM's MQSeries OS/390 middleware product in a batch or CICS environment. It reports the application CPU and wait time attributed to MQSeries calls, identifies the originating line of code, provides function descriptors for all MQSeries service modules, and summarizes information on the MQSeries call options and message attributes that the application uses.

STROBE and APMpower Online Library. The STROBE library of technical documentation in Adobe Acrobat PDF and IBM BookManager format. Users can view the STROBE and APMpower Online Library through Adobe Acrobat Reader or IBM Library Reader for Windows, IBM Library Reader/2, and BookManager READ/MVS.

STROBE Performance Profile. A hierarchical series of reports produced by STROBE that detail where and how time is spent during application execution.

STROBE Procedure. One of several job control procedures supplied with STROBE. These procedures can be customized and used instead of STROBE/ISPF to create reports.

STROBE Request Element (SRE). As each user request is received, STROBE checks the request for proper syntax and verifies that the submitter has access to the target address space. It then stores the measurement request as a STROBE Request Element (SRE).

STROBE Session Management Facility (SSMF). A STROBE component that controls measurement tasks within the address space of the target program or subsystem to collect performance data.

STROBE UNIX System Services Feature. A Compuware product that extends the functions of STROBE so that users can measure applications that run under OS/390 UNIX System Services. The STROBE Performance Profile reports are enhanced to identify system resource consumption and CPU activity and wait time for modules running in OS/390 UNIX System Services address spaces. The Feature reports on the use of Hierarchical File System (HFS) files and provides function descriptors for UNIX System Services modules.

STROBE-XCF Group. In a multisystem environment, STROBE uses the services of the MVS cross system coupling facility (XCF) for communication and configuration management. Each instance of STROBE in the sysplex has the option of joining the STROBE-XCF group, where it communicates with other members of the group through XCF signalling services.

Structured Query Language (SQL). Relational data language that uses English-like words for data

control, data definition, data manipulation, and query.

suspended request. A measurement session that is neither active nor completed. The session has been stopped by a user so that a task, such as switching sample data sets, can be performed. *(See also* active request, completed request, deferred request, measurement request, measurement session, queued request.*)*

sysplex. A set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.

SYSPRINT. The data definition name to which the listing from a compiler is written.

target job name. The name of the job that the user plans to measure, as specified in the JOBNAME field of the JCL JOB statement.

target job step. The job step of the application that is to be measured.

target sample size. The estimated number of samples the measurement task will take during the course of a measurement session.

task control block. The principle MVS control block used to control the execution of a task.

task wait time. All wait time not associated with file I/O or file management.

total time. The percentage of time that an activity used a resource. This percentage includes all CPU execution time and wait time. (*See also* CPU time, solo time, wait time.)

trigger request. One of the two types of requests in a *concurrent set*. A request group element that initiates and terminates the measurement sessions for all requests in a concurrent set. (*See also* concurrent set, related request.)

UNIX System Services Feature. *See* STROBE UNIX System Services Feature.

user-written data collector. A user-written program that STROBE invokes to supplement performance data collected by STROBE and its features or to collect performance data for environments STROBE does not support. Use of a data collector requires the STROBE Interface Feature. (*See also* STROBE data collector, STROBE Interface Feature.)

wait time. The portion of run time during which no task within the measured address space was able to make use of the CPU time available to it. (*See also* CPU time, run time, stretch time.)

XCF. Cross-system Coupling Facility.

Index

Special Characters

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X